

3100000011514



PERENCANAAN DAN PEMBUATAN WINDSHIELD WIPER OTOMATIS DENGAN FUZZY LOGIC

TUGAS AKHIR

PERPUSTAKAAN ITS 25-2-00	
Tgl. Terima	4
Terima Dari	8909
No. Agenda Prp.	

Disusun oleh :

ZUDHA NOORADILLAH

NRP : 2293.100.001

RSE
629.89
Noo
p-1
1998



JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1998

PERENCANAAN DAN PEMBUATAN WINDSHIELD WIPER OTOMATIS DENGAN FUZZY LOGIC

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro
Pada**

**Bidang Studi Elektronika
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
S u r a b a y a**

Mengetahui / Menyetujui

Dosen Pembimbing I



Ir. MURDI ASMOROADJI

NIP. 130 532 014

Dosen Pembimbing II



MOCH. RIVAI, ST. MT

NIP. 132 088 341

**S U R A B A Y A
Agustus, 1998**

ABSTRAK

Pesatnya perkembangan teknologi elektronika menunjang otomatisasi sistem kontrol, meskipun demikian masih ada kendala-kendala yang harus dipecahkan seperti sulitnya membuat kontrol otomatisasi pada sistem yang non linear dan dinamis. Penerapan teori *fuzzy* pada sistem kontrol merupakan teknologi baru yang menawarkan solusi dari permasalahan di atas. Dalam tugas akhir ini, telah direncanakan dan dibuat alat pengontrol pada *windshield wiper system* (pembersih kaca depan) pada kendaraan mobil dengan menggunakan *adaptive Fuzzy Logic Controller*.

Sensor konduktivitas digunakan untuk mendeteksi besarnya intensitas air hujan dan faktor kebasahannya, sedangkan sensor *optocoupler* untuk memantau posisi wiper. Selain itu digunakan juga *user timer* untuk memberikan aksi berupa pembersihan kaca secara periodik dan otomatis dalam waktu yang tertentu. Parameter-parameter tersebut digunakan sebagai masukan bagi kontroler untuk membuat *windshield wiper* menjadi otomatis.

Hasil pengujian membuktikan bahwa sistem ini dapat memberi kategori tanggapan berupa kecepatan motor wiper (sedang atau cepat) dan aktif atau tidaknya motor penyemprot air yang sesuai dengan kategori hujan berupa kondisi kebasahan (sangat kering, kering, kering kebasahan, basah, atau basah sekali) serta intensitas atau frekuensinya (tidak hujan, rintik-rintik, gerimis, deras, atau deras sekali).

Penerapan teknologi fuzzy ternyata sangat bermanfaat dalam bidang kontrol, memudahkan proses disain, dan mempunyai tingkat keandalan yang tinggi.

**“....Sesungguhnya tiada berputus asa
dari rahmat Alloh, melainkan kaum
yang kafir.”
(Qur'an surat Yusuf : 87)**

KUPERSEMBAHKAN
BUAT IBU, BAPAK, KAKAK, DAN ADIKKU
SEMOGA ALLOH SWT SELALU MEMBERIKAN
HIDAYAH DAN PERTOLONGAN-NYA KEPADA KITA

KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan kepada Alloh SWT atas segala hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir ini, dengan judul :

PERENCANAAN DAN PEMBUATAN WINDSHIELD WIPER

OTOMATIS DENGAN FUZZY LOGIC

Tugas akhir ini mempunyai beban 6 SKS (satuan kredit semester) yang disusun untuk memenuhi persyaratan menyelesaikan studi program S1 pada Jurusan Teknik Elektro Fakultas Teknologi Industri ITS Surabaya.

Dalam mengerjakan tugas akhir ini penulis mengacu pada teori-teori yang telah didapat selama kuliah, literatur, dan juga tidak lepas dari arahan dosen pembimbing yang telah banyak membantu dari awal hingga selesainya pengerjaan tugas akhir ini.

Ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada :

- Bapak Ir. Sutikno selaku Koordinator Bidang Studi Elektronika.
- Bapak Ir. Murdi Asmoroedji selaku Dosen Pembimbing I.
- Bapak Moch. Rivai, ST. MT selaku Dosen Pembimbing II.
- Bapak-bapak dosen, khususnya pada bidang studi Elektronika yang telah memberikan bekal ilmu selama penulis menempuh kuliah.
- Seluruh keluarga penulis atas segala bantuan, doa, dan dorongannya.

- Teman-teman di Bidang Studi Elektronika yang telah banyak memberikan sumbangan pemikiran serta dorongan kepada penulis dalam menyelesaikan tugas akhir ini.

Penulis berharap bahwa tugas akhir ini akan bermanfaat bagi pembaca serta dapat disempurnakan dan dikembangkan semaksimal mungkin.

Surabaya, Juli 1998

Penulis

DAFTAR ISI

	Halaman
Judul	i
Lembar Pengesahan	ii
Abstrak	iii
Kata Pengantar	iv
Daftar Isi	vi
Daftar Gambar	x
Daftar Tabel	xi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Permasalahan	2
1.3. Hipotesa	2
1.4. Tujuan	2
1.5. Metodologi	3
1.6. Relevansi	3
BAB II TEORI PENUNJANG	4
2.1. RANGKAIAN <i>OP-AMP</i>	4
2.1.1. Pengikut Tegangan (<i>Voltage Follower</i>)	4
2.1.2. Tapis Lulus Jalur (<i>Band Pass Filter</i>)	4
2.2. PENGUBAH A/D	5
2.2.1. Pendahuluan	5

2.2.2. Pengubah A/D dengan MAX174	7
2.3. MIKROPROSESSOR 8088	10
2.3.1. Organisasi memori 8088	10
2.3.2. Pengalamatan I/O	11
2.3.3. <i>Internal Register</i> Mikroprosesor 8088	11
2.3.4. Memori	12
2.3.5. <i>Programmable Peripheral Interface (PPI)</i> 8255	13
2.4. FUZZY LOGIC CONTROLLER	15
2.4.1. Deskripsi Umum	15
2.4.2. Karakteristik	16
2.4.3. Aplikasi	16
2.4.4. Deskripsi <i>Pin</i>	17
2.4.5. Arsitektur <i>Device</i>	19
2.4.6. <i>Membership Function</i> (MF)	21
2.4.7. Variabel <i>Fuzzy</i>	24
2.4.8. <i>Rule</i> (Aturan)	25
2.4.9. Evaluasi <i>Rule</i>	26
2.4.10. <i>Floating Membership Function</i>	26
2.4.11. <i>Operational Device</i>	27
2.5. Fotodioda	32

BAB III PEMBUATAN PERANGKAT LUNAK DAN PERANGKAT KERAS

34

3.1. PEMBUATAN SISTEM	34
3.2. PEMBUATAN PERANGKAT KERAS	36
3.2.1. Sensor Hujan	36
3.2.2. Sensor Posisi	37
3.2.3. <i>User Timer</i>	38
3.2.4. <i>Driver</i> Motor Wiper dan Semprot Air	40
3.2.5. Display Kerja Sistem	42
3.2.6. Modul NLX220	43
3.3. PEMBUATAN PERANGKAT LUNAK	44
3.3.1. Input	44
3.3.2. Output	45
3.3.3. Variabel <i>Fuzzy</i>	45
3.3.4. <i>Rules</i>	46
3.3.5. Simulasi dan <i>Download rule</i> ke IC	49
BAB IV PENGUKURAN DAN PENGUJIAN	50
4.1. Pengukuran Modul	50
4.1.1. Modul Sensor Hujan (Intensitas dan Kondisi Kebasahan)	51
4.1.2. Modul Sensor Posisi	52
4.1.3. Modul <i>User Timer</i>	52
4.1.4. Modul <i>Driver</i> Motor Wiper dan Semprot Air	52
4.1.5. Modul NLX220	53
4.2. Pengujian Sistem	54

BAB V PENUTUP	55
5.1. Kesimpulan	55
5.2. Saran	56
DAFTAR PUSTAKA	57
LAMPIRAN	

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Pengikut tegangan non inverting	4
Gambar 2.2. Tanggapan frekuensi dari sebuah <i>band pass filter</i>	5
Gambar 2.3. Diagram blok pengubah A/D jenis pencacah tanjakan	6
Gambar 2.4. Blok diagram NLX220	7
Gambar 2.5. Jenis Membership Function	22
Gambar 2.6. Membership Function Kecepatan	23
Gambar 2.7. Overlap Dua Membership Function	24
Gambar 2.8. Fuzzifikasi dari input Temperatur	25
Gambar 2.9. Floating Membership Function	26
Gambar 2.10. Mode immediate defuzzifikasi	31
Gambar 2.11. Mode accumulate defuzzifikasi	32
Gambar 3.1. Blok diagram sistem	35
Gambar 3.2. Rangkaian sensor hujan	37
Gambar 3.3. Rangkaian sensor posisi	38
Gambar 3.4. Rangkaian User Timer	40
Gambar 3.5. Rangkaian driver semprot air	41
Gambar 3.6. Rangkaian driver motor dc wiper	42
Gambar 3.7. Blok diagram display unjuk kerja sistem	42
Gambar 3.8. Rangkaian modul NLX220	43

DAFTAR TABEL

	Halaman
Tabel 2.1. Tabel kebenaran untuk MAX174	9
Tabel 2.2. Jenis Segment	10
Tabel 2.3. Absolute Maximum Rating Ta=25 °C	18
Tabel 2.4. Spesifikasi konversi analog	18

BAB I

PENDAHULUAN

1.1. Latar Belakang

Windshield wiper adalah bagian komponen mobil yang berfungsi sebagai pembersih kaca depan. Selama ini pengoperasian *windshield wiper* hanya dilakukan secara manual saja. Apabila dalam keadaan hujan pengemudi menyalakan tombol *windshield wiper*, sedangkan apabila hujan sudah berhenti maka pengemudi mematikan tombol secara manual.

Konfigurasi sistem *windshield wiper* terdiri dari motor wiper (untuk kecepatan sedang dan tinggi) serta motor semprot.

Untuk meningkatkan unjuk kerja dari *windshield wiper* maka metode yang ada sebelumnya dapat ditingkatkan dengan menggunakan teknologi *Fuzzy Logic*. Dengan ini diharapkan sistem *windshield wiper* ini dapat memonitor intensitas hujan sehingga alat dapat bekerja dengan otomatis dan pengemudi dapat lebih berkonsentrasi pada medan jalan yang dilalui.

Dasar-dasar *Fuzzy Logic* telah disinggung oleh banyak ilmuwan satu dua abad yang lampau sebagai jawaban atas lemahnya logika *Boolean*, namun baru pada sekitar pertengahan 1960-an Profesor Lotfi Zadeh menjabarkannya secara terperinci, logika ini ternyata mampu menjawab ketidakmampuan logika *Boolean* dalam mengenali perubahan parameter yang tidak jelas.

Pemakaian *Fuzzy Controller* adalah untuk menyederhanakan pengaturan. Biasanya sistem kontrol memakai implementasi persamaan matematis yang kompleks dalam hal pengaturan. Persamaan matematis yang demikian seringkali menemui kesulitan untuk diimplementasikan, karena begitu sangat berpengaruhnya ketidaklinearan *device* maupun ketergantungan di dalam lingkungan yang ditempati.

1.2. Permasalahan

Selama ini penggunaan sistem *windshield wiper* masih secara manual, sehingga diperlukan otomatisasi. Selain itu kecepatan pengeringannya konstan (tidak tergantung oleh keadaan hujan).

1.3. Hipotesa

Fuzzy Logic Controller sering digunakan sebagai pengaturan sistem secara adaptif sehingga sistem kontrol ini diharapkan dapat mengatasi permasalahan di atas.

1.4. Tujuan

Membuat perangkat keras dan lunak untuk membentuk prototipe suatu sistem pengaturan otomatis.

1.5. Metodologi

Metodologi pembuatan tugas akhir ini adalah melakukan studi literatur mengenai proses dan sistem *windshield wiper*.

Langkah berikutnya adalah mempelajari penggunaan *Fuzzy Logic* dalam proses kontrol suatu sistem. Dalam hal ini mempelajari juga *chip NLX220*.

Selanjutnya dilakukan perencanaan dan pembuatan perangkat keras berupa sistem untuk mengontrol sistem *windshield wiper*.

Pembuatan perangkat lunak merupakan langkah berikutnya yang berupa pembuatan *rule-rule* untuk *Fuzzy*.

Langkah selanjutnya adalah melakukan pengujian dan pengukuran alat dimana alat diuji perbagian dahulu baru diuji secara keseluruhan sistem.

Setelah selesai maka seluruh langkah di atas disusun dalam suatu laporan tugas akhir.

1.6. Relevansi

Dari tugas akhir ini diharapkan dapat menghasilkan referensi dan masukan bagi industri otomotif yang ingin meningkatkan unjuk kerja *windshield wiper* dengan memakai teknologi *Fuzzy Logic* dan prototipe yang dihasilkan dapat diaplikasikan pada sistem *windshield wiper* pada industri otomotif.

BAB II

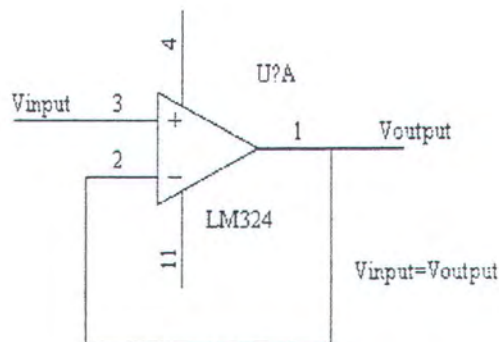
TEORI PENUNJANG

2.1. RANGKAIAN OP-AMP ¹

2.1.1. Pengikut Tegangan (*Voltage Follower*)

Pengikut tegangan biasanya didefinisikan sebagai rangkaian dengan penguatan satu dengan keluaran mengikuti masukan.

Dengan pengikut tegangan *non inverting* (Gambar 2.1), keluaran terhubung langsung pada masukan membalik dengan tegangan masukan dikenakan pada masukan *non inverting*. Resistansi umpan balik sama dengan nol; karena itu, sesuai penguatan tahapan untuk tak membalik, atau penguat pengikut sama dengan 1.

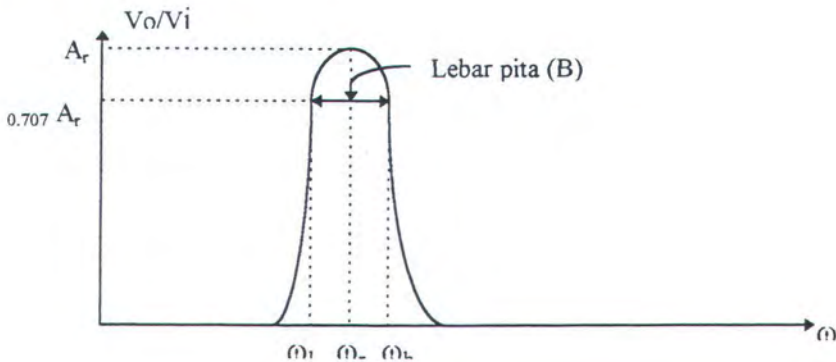


Gambar 2.1. Pengikut tegangan *non inverting*.

¹ Penguat Operasional dan Rangkaian Terpadu Linear

2.1.2. Tapis Lulus Jalur (*Band Pass Filter*)

Band pass filter adalah sebuah rangkaian yang dirancang untuk hanya melewatkan isyarat dalam suatu pita frekuensi tertentu seraya menolak semua isyarat di luar pita ini.



Gambar 2.2. Tanggapan frekuensi dari sebuah *band pass filter*

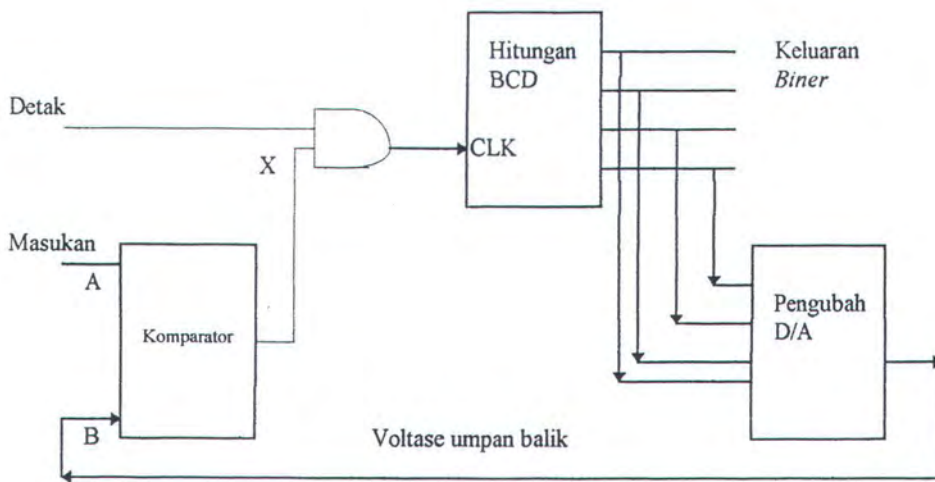
Filter-filter ini digolongkan sebagai pita sempit atau pita lebar. Filter pita sempit adalah sebuah filter yang mempunyai lebar pita lebih kecil atau sepersepuluh frekuensi resonannya. Jika lebar pitanya lebih besar daripada sepersepuluh frekuensi resonannya, filter tersebut merupakan sebuah filter pita lebar.

2.2. PENGUBAH A/D

2.2.1. Pendahuluan

Pengubah analog ke digital merupakan jenis khusus dari pengkode. Diagram blok dasar dari pengubah A/D ditunjukkan pada gambar 2.3. Masukannya berupa tegangan berubah tunggal . Tegangan pada kasus ini berubah

dari 0 sampai 3 V. Keluaran pengubah A/D berupa *biner*. Pengubah A/D menerjemahkan tegangan analog pada masukan ke dalam kata *biner* empat bit. Seperti pada pengkode lain, kita dapat menentukan dengan tepat masukan dan keluaran yang kita harapkan.



Gambar 2.3. Diagram blok pengubah A/D jenis pencacah tanjakan

Pada gambar tersebut dimisalkan terdapat suatu logika 1 pada titik X dikeluarkan pembanding. Dianggap pula pencacah BCD ada pada biner 0000 dan 0.55 V terpasang pada masukan analog. Logika 1 pada X membuka gerbang AND dan pulsa pertama dari detak muncul pada masukan CLK dari pencacah BCD. Pencacah melanjutkan cacahannya sampai 0001 yang ditampilkan pada keluaran biner dan dimasukkan ke pengubah D/A. Misal pada pengubah D/A mempunyai kenaikan 0.2 V tiap perubahan bit maka 0001 diubah menjadi 0.2 V yang dimasukkan pada titik B dari pembanding. Pembanding memeriksa masukannya. Masukan A lebih tinggi (0.55 V dibanding dengan 0.2 V), dengan demikian

maka pembanding mengeluarkan logika 1 yang kemudian membuka kembali gerbang AND dan pulsa detak melewati pencacah. Pencacah melanjutkan cacahannya dengan satu menjadi 0010 dan masuk ke pengubah D/A menjadi 0.4 Volt. Pembanding masih mengeluarkan logika 1 dan pencacah menjadi 0011 yang kemudian dimasukkan ke pengubah D/A menjadi 0.6 V. Pembanding selanjutnya memeriksa masukan A dan B; untuk pertama kalinya masukan B lebih besar daripada masukan A. Pembanding mengeluarkan logika 0 yang menyebabkan pencacah menghentikan cacahannya pada biner 0011.

2.2.2. Pengubah A/D dengan MAX174

MAX174 merupakan pengubah A/D 12 bit yang mempunyai kecepatan tinggi, konsumsi daya yang rendah, *clock* yang sudah terpasang (*on chip*), dan referensi tegangan. MAX174 memerlukan waktu konversi $8\mu\text{s}$ dan komponen eksternal yang terbatas pada kapasitor dan resistor yang harganya sudah tertentu.

MAX174 menggunakan standar *interface* mikroprosesor yang dapat dihubungkan ke 8, 12, dan 16 lebar *bus*. Tiga kondisi data output dikontrol oleh logika input $\overline{\text{CS}}$, CE, dan $\text{R}/\overline{\text{C}}$.

Proses konversi

MAX174 menggunakan cara *successive approximation* untuk mengubah input analog menjadi 12 bit kode output digital. Kontrol logika menyediakan *interface* yang mudah pada mikroprosesor. Sebagian besar aplikasi menggunakan komponen pasif secara eksternal untuk menampilkan fungsi analog ke digital.

Tegangan output internal D/A dikontrol oleh *Successive Approximation Register* (SAR) dan mempunyai output impedansi sebesar $2.5k\Omega$. Input analog dihubungkan output D/A dengan resistor $5k\Omega$ untuk input 10 V dan resistor $10k\Omega$ untuk input 20 V. Komparator atau pembanding yang digunakan menggunakan *zero crossing detector* dan outputnya diumpanbalikkan ke input SAR.

Pada saat konversi dimulai, SAR diset pada setengah skala. Input analog dibandingkan dengan setengah dari tegangan skala penuh. Hasilnya berlogika 1 bila input analog lebih besar daripada setengah skala atau berlogika 0 bila lebih kecil. Pada akhir konversi, output SAR disimpan pada *output buffers* atau penyangga.

Digital Interface

\overline{CS} , CE, dan R/\overline{C} mengontrol proses secara keseluruhan. Apabila CE dan \overline{CS} digunakan, keadaan R/\overline{C} dipilih logika 0 jika pada keadaan konversi atau logika 1 jika akan membaca data yang sudah dikonversi. A0 biasanya disesuaikan dengan LSB dari *address bus*. Untuk menampilkan konversi 12 bit secara penuh dapat dilakukan dengan mengeset A0 *low* selama proses konversi. Sedangkan untuk konversi 8 bit, A0 harus diset *high* selama proses konversi.

Format data output

Selama proses pembacaan data, A0 juga menyeleksi apakah *three state buffers* berisi 8 *MSBs* (A0=0) atau 4 *LSBs* (A0=1) sebagai hasilnya. 4 *LSBs* diikuti dengan 4 *trailing* 0s.

Tabel 2.1. Tabel kebenaran untuk MAX174

CE	$\overline{\text{CS}}$	R/ $\overline{\text{C}}$	12/ $\overline{8}$	A0	KETERANGAN
0	X	X	X	X	Tidak ada operasi
X	1	X	X	X	Tidak ada operasi
1	0	0	X	0	Inisialisasi konversi 12 bit
1	0	0	X	1	Inisialisasi konversi 8 bit
1	0	1	1	X	Mengaktifkan output paralel 12 bit
1	0	1	0	0	Mengaktifkan 8 <i>MSBs</i>
1	0	1	0	1	Mengaktifkan 4 <i>LSBs</i> + 4 <i>trailing</i> 0s

Output data yang dihasilkan berdasarkan pin 12/ $\overline{8}$. Jika diberi masukan *low* maka output akan dibagi menjadi dua 8 bit *bytes*. Hal ini menyebabkan *interface* 8 bit tanpa membutuhkan *three states buffers* eksternal. Jika diberi masukan *high* maka output akan menjadi 12 bit *word*.

Untuk memulai konversi, mikroprosesor harus menulis alamat *ADC*. Pada proses konversi biasanya memerlukan waktu yang lebih lama daripada *single clock cycle*, maka mikroprosesor harus menunggu *ADC* untuk menyelesaikan proses konversinya. Hasil data yang diambil akurat apabila sudah pada akhir konversi yang ditandai dengan *STS*.

Setelah proses konversi selesai, mikroprosesor dapat mengambil hasilnya dan diolah sesuai dengan keinginan.

2.3. MIKROPROSESSOR 8088

2.3.1. Organisasi memori 8088

Mikroprosesor 8088 memiliki 20 bit alamat ke memori. Memori pada 8088 diorganisasi sebagai *array* yang berdimensi satu, sebesar 1 *Mega byte*, dan dialamati dari 00000H sampai FFFFFH. Secara logika, memori dibagi dalam *Code*, *Data*, dan *Stack Segment* yang masing-masing besarnya maksimum 64 *KByte*.

Semua referensi dalam memori dibuat relatif terhadap *Segment Register*. Penggunaan *Segment Register* dipilih secara otomatis oleh prosesor sesuai dengan tabel 2.2 . *Operand Word* (16 bit) dapat diletakkan pada alamat ganjil maupun alamat genap dengan urutan *byte* bawah ditaruh pada alamat pertama dan *byte* atas ditaruh pada alamat selanjutnya.

Tabel 2.2. Jenis *Segment*

Referensi Memori	<i>Segment Register</i>
Instruksi	<i>Code</i> (CS)
<i>Stack</i>	<i>Stack</i> (SS)
<i>Local Data</i>	<i>Data</i> (DS)
<i>Eksternal Global Data</i>	<i>Extra</i> (ES)

Lokasi memori tertentu digunakan untuk operasi *CPU* yang spesifik. Lokasi dari FFFF0H sampai dengan FFFFFH digunakan untuk meletakkan operasi *Jump* ke awal program. Jika *CPU* di reset maka *CPU* akan melakukan instruksi yang terletak di alamat 003FFH, dapat digunakan untuk vektor tabel jika dilakukan operasi *interrupt* pada prosesor 8088.

2.3.2. Pengalamatan I/O

Pada 8088, operasi *I/O* dapat dialamati sampai 64 KByte *I/O register*.

Format alamat *I/O*, sama dengan alamat memori pada jalur *bus A0* sampai dengan *A15*. Jalur *A16* sampai dengan *A19* berharga 0 pada saat operasi *I/O*.

2.3.3. Internal Register Mikroprosesor 8088

Mikroprosesor 8088 memiliki lima kelompok register 16 bit, yaitu *Data register*, *Index register*, *Pointer register*, *Segment register* dan *Flag register*.

Empat register Data : *AX*, *BX*, *CX*, *DX*, dapat digunakan sebagai sebuah register 16 bit atau sebagai dua register 8 bit terpisah, misalkan register *AX* (16 bit) dapat digunakan dalam bentuk register *AH* (8 bit *High Byte*) dan *AL* (8 bit *Low Byte*).

Lima *Indeks* dan *Pointer register* mempunyai fungsi spesifik seperti :

1. *SP(Stack Pointer)* adalah register sebagai penunjuk lokasi puncak *stack*.
2. *BP(Base Pointer)* digunakan untuk menunjukan suatu lokasi di dalam *stack*.
3. *SI(Source Index)* digunakan sebagai *source index register* pada *mode indirect addressing*, juga digunakan untuk menyimpan *offset* suatu *operand* dalam berlangsungnya suatu operasi *string*.
4. *IP(Instruction Pointer)*, digunakan sebagai penunjukan dari alamat instruksi yang akan di eksekusi.

Empat *Segment Register* : *CS*, *DS*, *SS* dan *ES* sebagai *base* untuk ke empat *segment* memori : *Code Segment*. Lokasi memori absolut diperoleh dengan menggeser register tersebut 4 bit kekiri, kemudian ditambahkan suatu nilai *offset*.

Offset ini dapat berupa suatu konstanta atau register lain, misalnya register *BX*, *SI*, *DI* atau *IP*.

2.3.4. Memori

Memori dapat dibagi menjadi dua golongan, yaitu : *ROM* (*Read Only Memory*) dan *RAM* (*Random Access Memory*). Memori jenis *ROM* bersifat *Non-Volatile*, yang artinya data yang disimpan tidak akan hilang ketika hubungan *power supply* ke *ROM* terputus. Jenis *ROM* yang banyak digunakan adalah *EPROM* (*Erasable Programmable Read Only Memory*).

EPROM mempunyai jalur *address* sesuai dengan kapasitas memorinya. Sebagai contoh, *EPROM 2764* mempunyai kapasitas 8 Kilobyte ($8192 \times 8 = 2^{13} \times 8$), mempunyai 13 jalur *address* yaitu : *A0* sampai dengan *A12*. Pada *EPROM* terdapat pin *CS* (*Chip Select*) yang berfungsi mengaktifkan rangkaian *buffer I/O internal* dari *EPROM*. Pada *EPROM* ini juga terdapat pin *OE* (*Output Enable*) yang berfungsi mengaktifkan data output dari *EPROM*.

Jalur *address* dari *CPU* yang tidak dihubungkan ke *EPROM* disebut *zone bit*. Untuk *EPROM 2764* yang mempunyai interval memori *A0 - A12* mempunyai *zone bit A13 - A19*, bila dihubungkan dengan mikroprosesor 8088.

Untuk memori jenis *RAM* digunakan untuk menyimpan data untuk sementara waktu. Data pada *RAM* akan hilang bila *power supply* terputus. Ada dua jenis *RAM* ,yaitu *RAM Dinamis* dan *RAM Statis*. *RAM Statis* mempunyai keuntungan dalam hal kemudahan pengoperasiannya, karena tidak memerlukan *re-fresh*. Tetapi kerugiannya adalah harga dan ukuran perbit lebih besar dari pada

RAM Dinamis. Pada *RAM*, selain terdapat *pin CS (Chip Select)* dan *OE (Output Enable)* seperti pada *EPROM*, juga terdapat *pin WE (Write Enable)*.

2.3.5. Programmable Peripheral Interface (PPI) 8255

PPI 8255 merupakan *peripheral interface* yang dapat diprogram fungsinya. PPI ini dibagi menjadi dua group, yaitu group A dan group B, dimana group A terdiri dari *Port A* dan *Port C Upper*, sedangkan group B terdiri dari *Port B* dan *Port C Lower*. Fungsi dari *pin* pada PPI 8255 adalah :

Data (D0 - D7)	: Merupakan jalur data.
Chip Select (CS)	: Untuk mengaktifkan dan tidak mengaktifkan PPI 8255.
Read (RD)	: Untuk membaca data.
Address (A0 - A1)	: Menentukan <i>internal register</i> PPI 8255 yang akan dibaca atau ditulis.
Reset	: Untuk mereset <i>register internal</i> PPI 8255, setelah terjadi <i>reset</i> , semua <i>I/O Port</i> di-set dalam mode input.
Port A (PA0 - PA7)	: Digunakan untuk menghubungkan PPI 8255 dengan peralatan luar.
Port B (PB0 - PB7)	: Mempunyai fungsi yang sama dengan <i>Port A</i> .
Port C (PC0 - PC7)	: Mempunyai fungsi yang sama dengan kedua <i>Port</i> tersebut diatas. <i>Port C</i> dibagi menjadi dua bagian

yaitu *Port C Lower (PC0 - PC3)* dan *Port C Upper (PC4 - PC7)*.

Control Word register berfungsi untuk menentukan fungsi dari setiap *Port* dan menentukan *mode* PPI yang digunakan. PPI 8255 mempunyai tiga mode yang dapat dioperasikan, yaitu : *Mode 0 (Basic Input/Output)*, *Mode 1 (Strobe Input / Output)* dan *Mode 2 (Bidirectional Bus)*.

Mode 0 (Basic Input / Output). Mode ini digunakan untuk operasi *Input/Output* yang sederhana yang terdiri dari tiga *Port*. Tidak ada *Handshaking*, dan data yang dibaca dan ditulis pada *port-port* yang bersangkutan. Fungsi dasar *Mode 0* yaitu : Terdiri dari 2 *port* 8 bit dan 2 *Port* 4 bit. *Port-port* tersebut bisa sebagai *Input* atau *Output*. *Output* data di- *Latch* dan *Input* data tidak di-*Latch*. Terdapat 16 konfigurasi *Input / Output* yang mungkin.

Mode 1 (Strobe I/O). Konfigurasi pada operasi *Mode 1* menyediakan fasilitas untuk transfer data *I/O* dari dan ke *Port* tertentu dengan dilengkapi sinyal *Handshaking*. Dalam hal ini *Port A* dan *Port B* dapat digunakan untuk transfer data, sedangkan *Port C* sebagai pembangkit sinyal *Handshaking*.

Mode 2 (strobe Bidirectional I/O). Konfigurasi operasi ini menyediakan fasilitas untuk komunikasi data 8 bit dua arah dengan peralatan luar. Tersedia sinyal-sinyal untuk *Handshaking* dan *Interrupt* dengan fungsi *Enable* dan *Disable*-nya.

Sebelum melakukan operasi PPI 8255 pada *Mode 0* maka harus dilakukan inisialisasi pada PPI dengan mengirim sinyal *Control Word*.

2.4. FUZZY LOGIC CONTROLLER ²

2.4.1. Deskripsi Umum

NLX220 merupakan *device* yang membentuk kalkulasi logika *fuzzy* secara langsung pada perangkat keras. Karena memang dibuat untuk kontroller, maka mudah dalam pemakaian, performansi, dan tangguh dalam lingkungan yang kasar.

Device ini terdiri dari 4 analog input dan output dengan sumber *internal clock*. NLX220 akan menyerap daya yang rendah saat operasi normal dan mempunyai *mode power down* yang akan mengurangi daya dengan faktor 10.

Fuzzy logic sangat sesuai dengan proses-proses yang mempunyai input data yang acak dan sistem tidak linier untuk laju sistem kontrol yang tangguh.

Metodologinya memakai deskripsi secara linguistik dari sistem, sehingga menjadikannya sangat intuitif dan mudah untuk dipakai. Dapat juga dipakai untuk menambahkan kecerdasan pada produk-produk industri, misalnya untuk meningkatkan unjuk kerja, dan meningkatkan efisiensi.

NLX220P bisa diprogram yang sesuai untuk ruang kerja dan produksi yang terbatas. Kompatibilitas *pin* NLX220 memakai teknologi OTP (*One Time Programming*) untuk *storage* dan sesuai untuk produksi yang beragam.

Memori menyimpan MF (*Membership Function*) Fuzzy dan *parameter rule*. Pengorganisasian memori fleksibel dan dengan efisien mengadaptasi keperluan dari aplikasinya. *Device* ini menyimpan 111 variabel Fuzzy yang diorganisasikan dalam bentuk keperluan *rule*-nya.

² ----, *Stand Alone Fuzzy Logic Controllers NLX220*, Adaptive Logic.

Device menyediakan 6 tipe MF yang berbeda untuk berbagai aplikasi. MF mempunyai kemiringan konstan dan hanya perlu spesifikasi tipe, lebar, dan center.

NLX220 juga menyediakan *floating* MF, dimana lebar dan center bisa ‘float’ dibuat berubah-ubah secara dinamis. *Floating* MF dimanfaatkan untuk mengukur penurunan, membuat *timer*, atau meng-*adjust* untuk men-*drive* sensor.

Ada dua metode *Deffuzzifikasi*, *immediate* dan *accumulate*. *Immediate* akan men-*drive* output untuk harga yang sudah tertentu dan *accumulate* untuk menambahkan harga yang telah ada.

2.4.2. Karakteristik

NLX220 merupakan sebuah IC *Contoller Fuzzy Logic* yang lengkap yang mempunyai fleksibilitas dan *self adapting* yang tinggi.

Di dalamnya terdapat EEPROM atau OTP untuk memasukkan *rule-rule* yang telah dibuat.

Selain itu NLX220 mempunyai empat bit input analog, empat bit output analog, enam tipe MF, 111 variabel *fuzzy*, serta 50 *rules*.

2.4.3. Aplikasi

Aplikasi NLX220 diantaranya untuk *power* dan manajemen baterai, kontrol motor, kontrol pemanas, otomotif, serta industri kontrol.

2.4.4. Deskripsi Pin

Input

RESET, untuk menginisialisasi *device* dengan sinyal aktif *low*. Harus tetap aktif hingga sedikitnya 8 *clock* untuk memastikan operasi yang lama telah habis. Dapat diaktifkan dengan rangkaian *delay power-up*. Dengan Reset akan mengaktifkan mode *low-power*.

AIN(0-3), input data analog yang dengan internal akan dikonversikan ke 8 bit data digital. Input yang tidak dipakai harus di-*ground*-kan.

XIN, *clock input*, boleh dipakai eksternal input clock atau dengan kristal, di mana ujung satunya di-*ground*-kan.

PROG, untuk saat pemrograman NLX220P. *Pin* ini tidak dipakai pada NLX220. Saat operasi harus di-*ground*-kan.

PRESCALE, input logika '1' menandakan dalam mode *prescale* dan '0' dalam operasi normal. *Pin* ini di-*ground*-kan saat mode *prescale* tidak pernah digunakan atau dihubungkan dengan *pin* READY untuk operasi kontinyu. Mode juga bisa dipanggil selama pengoperasian oleh logika eksternal. Setelah RESET diaktifkan, PRESCALE input harus dipertahankan pada logika rendah sedikitnya selama 4 clock.

Output

AOUT (0-3), Analog output, 8 bit data digital dikonversikan secara internal ke level analog.

READY, setelah reset *pin* ini menandakan device mulai men-*sample* dan memproses data. *Pin* ini seharusnya tidak dihubungkan atau disambungkan dengan PRESCALE selama pengoperasian.

VREF, mem-*filter* referensi tegangan internal, hubungkan ke ground dengan 0,1uF kapasitor.

Tabel 2.3. *Absolute maximum Ratings* Ta = 25 °C

Parameter	Minimum	Maksimum	Unit
Vdd	-0,5	7,0	V
Vss	0	0	V
Input Digital	0	Vdd	V
Input Analog	0	Vdd	V
<i>Power Dissipation</i>		100	mV
<i>Storage temperatur</i>	-50	150	°C

Tabel 2.4. Spesifikasi konversi analog

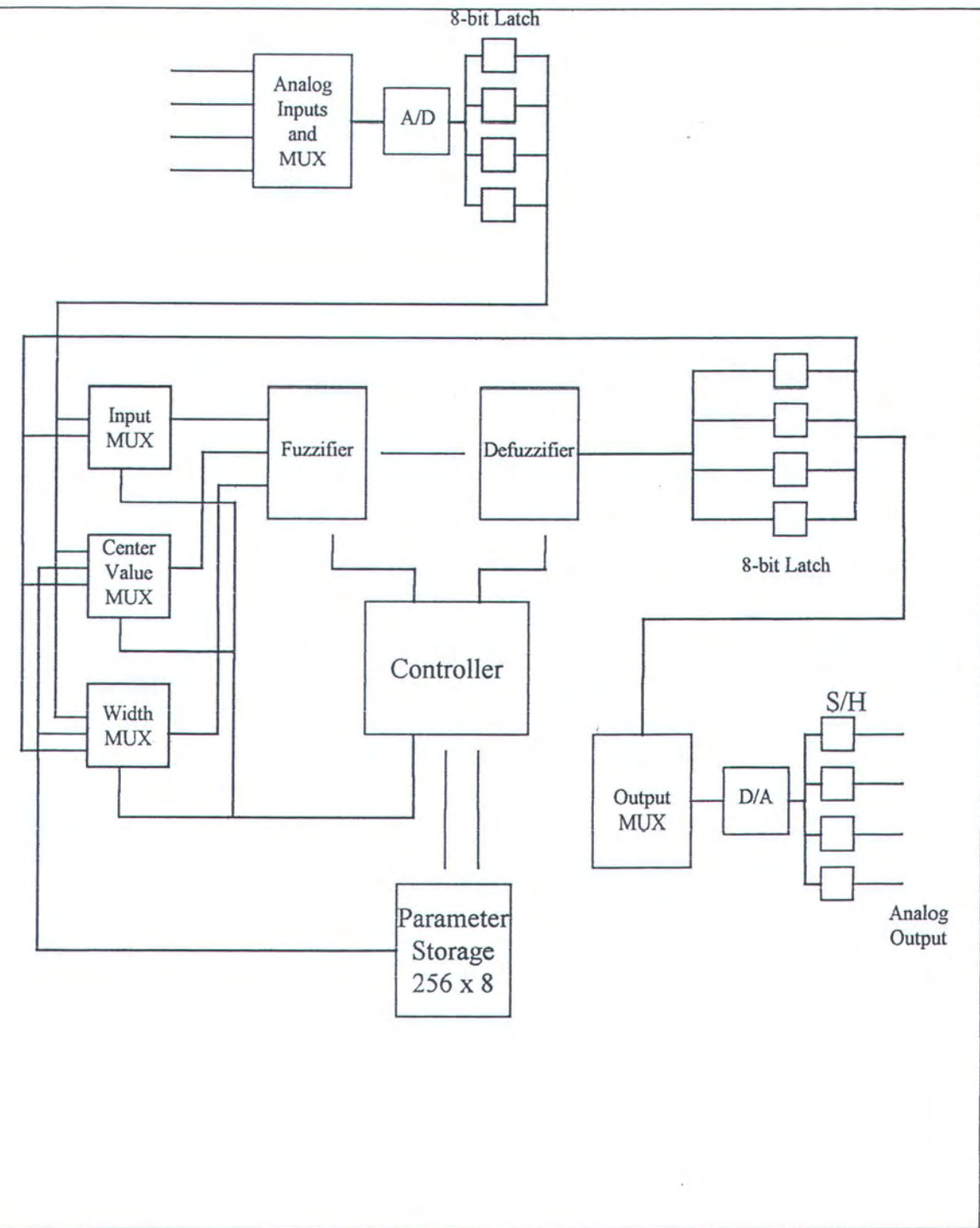
Parameter	Harga	Unit
Resolusi	1	Bit
<i>Slew Rate, Tracking</i>	1,6	V/ms max
<i>Zero Code Error</i>	1x	LSB
<i>Full Scale Error</i>	1x	LSB
<i>Signal to Noise Ratio</i>	45	dBmin
<i>Sampling Rate</i>	10KHz	Per Channel

2.4.5. Arsitektur *Device*

Device ini adalah *stand alone Controller Fuzzy Logic* yang membentuk semua kalkulasi di dalam *hardware* dan tidak memerlukan *software*. Input dapat secara langsung dihubungkan ke sensor atau *switch*, demikian juga outputnya langsung dihubungkan dengan piranti analog atau digunakan untuk fungsi kontrol.

Komponen utama NLX220 adalah *Fuzzifier*, *Deffuzzifier*, dan *Kontroller*. *Fuzzifier* mengkonversikan input data ke dalam data Fuzzy, dan dalam hubungannya dengan *kontroller*, akan mengevaluasi data fuzzy dengan definisi set *rule* yang dimasukkan yang menggambarkan sistem kontrol yang dimaksud. Setelah *rule-rule* dievaluasi, *Deffuzzifier* memberikan nilai aksi ke output yang bersesuaian.

Gambar Blok diagram NLX220 seperti berikut :



Gambar 2.4. Gambar Blok diagram NLX220

2.4.6. *Membership Function (MF)*

MF dipakai untuk membagi input ke dalam bagian-bagian dimana inputnya biasanya bervariasi. MF dibandingkan dengan data input untuk mengetahui dimana data tersebut akan ditempatkan. Tempat-tempat tersebut tergantung disainernya dalam mengklasifikasikan data, misalnya hangat, cepat, atau tinggi.

Dalam hal ini termometer, pembagian suhunya dibuat sehalus mungkin, misal :

1. Di bawah 60 F = Cold
2. 60 F - 70 F = Cool
3. 70 F - 75 F = Moderat
4. 75 F - 85 F = Warm
5. Di atas 85 F = Hot

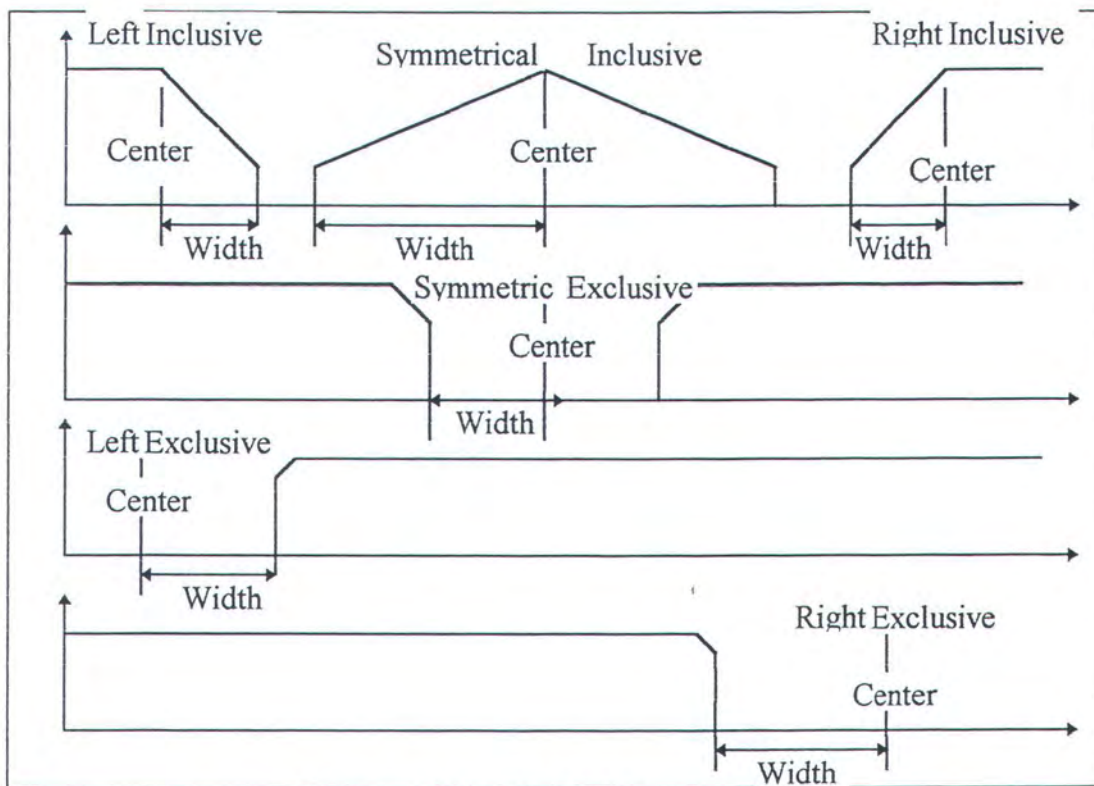
Pembagian ini hanya secara intuitif saja. Di dalam Fuzzy Logic 5 bagian ini disebut MF. Pembagian ini boleh terjadi *overlap*, dimana datanya berarti anggota dari kedua MF. Misalnya dingin dengan cold.

NLX220 mempunyai 6 macam *slope*:

1. *Left Inclusive*
2. *Symmetrical Inclusive*
3. *Right Inclusive*
4. *Symmetrical Exclusive*
5. *Left Exclusive*
6. *Right Exclusive*

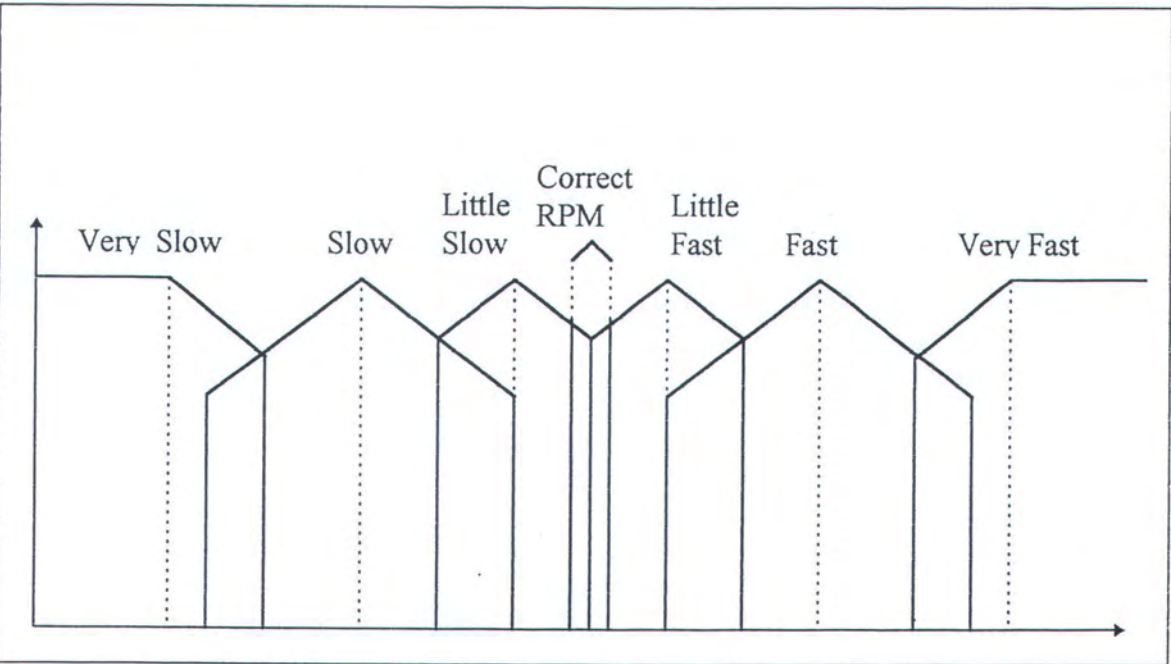
Di dalam aplikasinya didefinisikan dengan nama, tipe bentukannya, dan nilai numerik *center* dan *width*-nya. Pemilihan MF harus hati-hati agar dapat menyederhanakan banyak model. Misalnya, dalam termometer, Dingin adalah *Left inclusive* dan Panas *Right Inclusive* MF.

Gambar jenis *Membership Function*

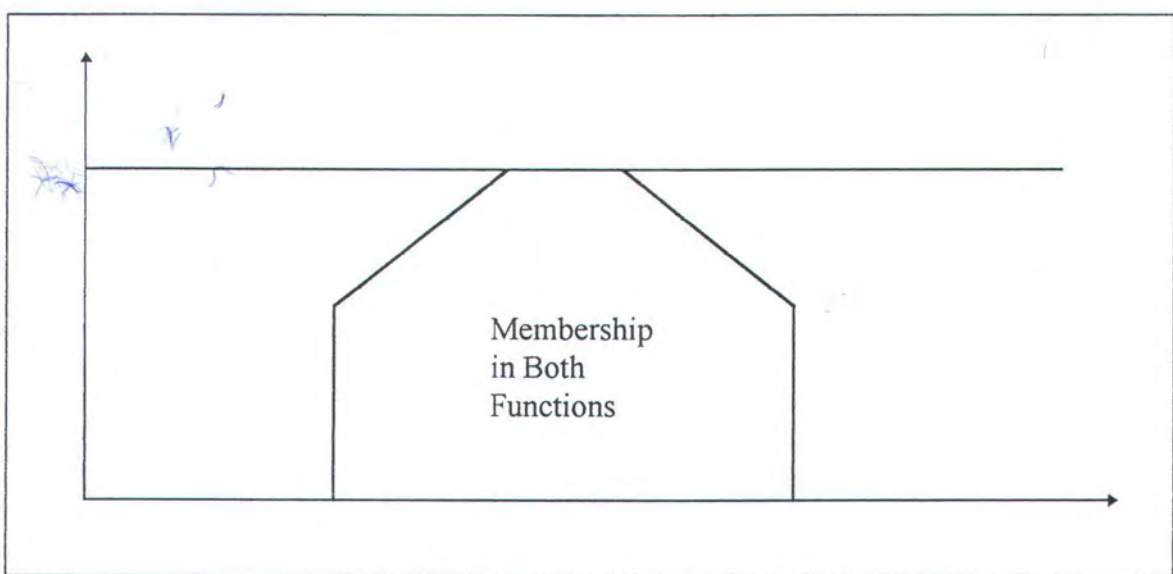


Gambar 2.5. Jenis *Membership Function*

Ketepatan kontrol pada *operating point* yang diinginkan dapat diberikan dengan sempitnya *Symmetrical Inclusive MF*. Aplikasinya kontrol motor, yang perlu sekali ketepatan. Contoh dari gabungan dari tipe dan lebar yang berbeda dipakai untuk memonitor kecepatan motor.

Gambar *Membership Function Kecepatan*Gambar 2.6. *Membership Function Kecepatan*

MF dapat di-*overlap*-kan agar membentuk tipe MF baru seperti *trapezoidal*, yang merupakan gabungan dari *Left Inclusive* dan *Right Inclusive*. Data input yang masuk ke dalam tipe *trapezoid* adalah anggota dari kedua MF tersebut.



Gambar 2.7. *Overlap Dua Membership Function*

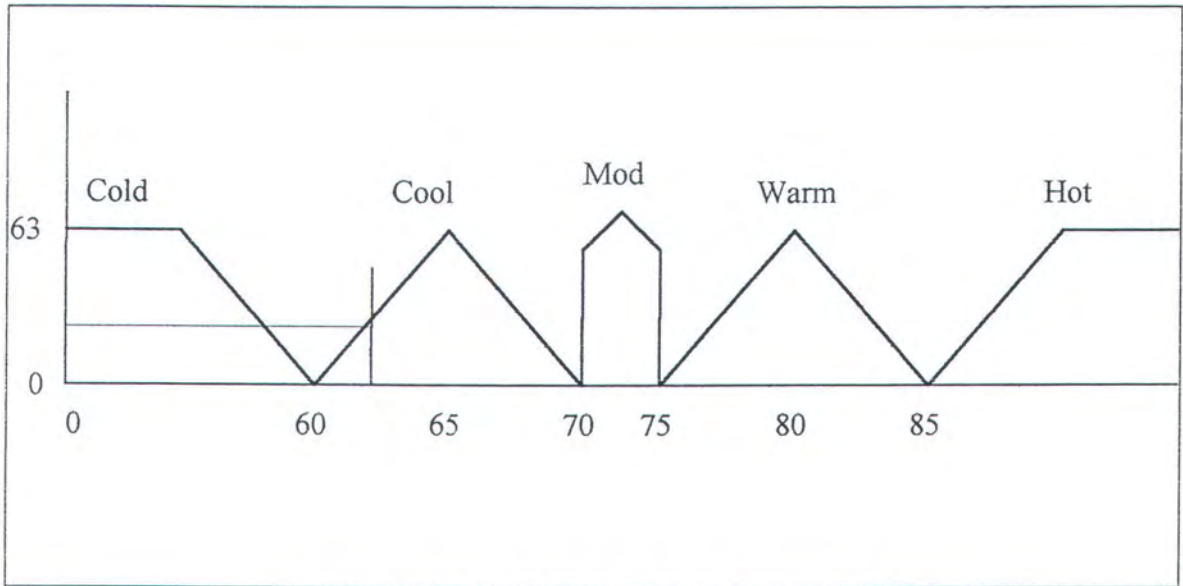
2.4.7. Variabel *Fuzzy*

Adalah ekspresi linguistik yang menunjukkan input bersesuaian dengan MF di sumbu mendatarnya. Variabel Fuzzy berdasarkan pada *Membership Function* dan Input variabel, seperti misalnya :

if Temperatur is Cool

Di dalam contoh ini 'Temperatur' adalah input dan 'Cool' adalah *Membership Function*.

Hubungannya dikerjakan oleh *Fuzzifier*, hasilnya adalah data *Fuzzy* yang menunjukkan derajat mana data input yang sesuai dengan MF. Data *Fuzzy* adalah numerik dan berkisar antara 0 - 63 di dalam NLX220.



Gambar 2.8. Fuzzifikasi dari input Temperatur

2.4.8. Rule (Aturan)

Rule adalah berisi satu atau lebih variabel Fuzzy dan sebuah nilai aksi ke outputnya. *Rule* dipakai untuk memberitahu ke kontroller bagaimana menanggapi perubahan input data.

Misalnya :

Output -5 if Velocity is Fast and Acceleration is Positive

Output +5 if Velocity is Little_Slow and Acceleration is Zero

Di *rule* pertama, variabelnya adalah 'Velocity is Fast' dan kedua 'Acceleration is Positive'. Aksi '-5' dan '+5' diberikan ke output untuk mengurangi atau mempercepat motor. Jika memakai tanda ' \pm ' berarti memakai mode output *accumulate* yang menunjukkan bahwa output bisa ditambah atau dikurangi.

2.4.9. Evaluasi Rule

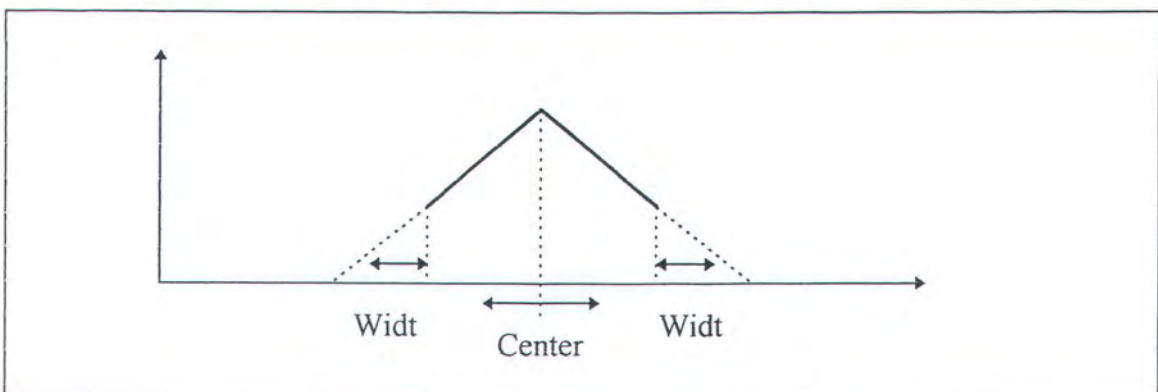
Ada beberapa metode untuk mengevaluasi *Rule Fuzzy Logic NLX220* mengevaluasi dengan teknik dua step MAX-of-MIN.

Step pertama - MIN, semua nilai variabel Fuzzy dibandingkan dan nilai paling rendah mewakili *Rule*. Step kedua - MAX, nilai *rule* dibandingkan dan nilai paling tinggi yang menang.

Membership function, variabel Fuzzy, dan *Rule* dibuat dan dikelompokkan menurut keperluan aplikasi. Sifat-sifat fisik sistem yang mau dikontrol harus dipahami sebelum memasukkan model Fuzzy.

2.4.10. Floating Membership Function

Keistimewaannya memakai fungsi *Floating MF*. *Floating* yang dimaksudkan adalah nilai *center* dan *width* dari MF dapat dibuat berubah-ubah, yang biasanya adalah nilainya tetap dan disimpan di memori. Di dalam *floating membership function* nilainya dapat berasal dari input atau output.



Gambar 2.9. *Floating Membership function*.

Beberapa MF dibuat floating saat memasukkan data. *Floating* MF berfungsi merubah nilai *center* dan *width* sebagai data dari perubahan pilihan input atau output.

Misalnya :

IN1 is small (0, 25, Symmetrical Inclusive)

IN2 is small (0, 25, Symmetrical Inclusive)

Dimana : 0 = *center*

 25 = *width*

Dua variabel Fuzzy tadi dapat digabungkan menjadi:

Output +1 if IN 1 is small and IN2 is small

dimana varabel fuzzy 'IN1 is small' membandingkan input IN1 dengan membership function konvensional 'small'. *Floating* MF membuatnya akan menjadi lebih ringkas dengan variabel Fuzzy dan rule berikut :

IN1 is small_difference (IN2, 25, symmetrical Exclusive)

Output +1 if IN1 is small_difference

Di dalam variabel Fuzzy, *center* dari MF small_difference didefinisikan oleh nilai IN2 yang disimpan di *latch input*.

Saat proses *Fuzzifikasi*, sebuah input dikurangkan dari *center* dan nilai absolutnya dikonversikan untuk mengukur bagaimana sedekat mungkin hal itu dapat *match* dengan nilai *centernya*. Ketika *fuzzifikasi*, *floating* MF akan mengurangi satu input dengan yang lain.

Floating MF seperti contoh di atas digunakan untuk mengkalibrasi input sensor *over time*, dengan cara langsung membandingkan dua input. Nilai stabil

sensor dibandingkan dengan set tegangan. *Rule* kalibrasi mengecek derajat dari ketidaktepatan dan menyimpannya ke dalam *output latch*. Jika input dalam kalibrasi, *center* akan *match* dan nilai koreksi adalah nol. Koreksi ketidaktepatan yang besar akan menyimpan nilai yang besar juga. Koreksi digunakan untuk meng-*adjust floating center* dari MF di dalam *rule* yang memproses data sensor.

Floating MF dapat digabungkan dengan aksi *floating output* untuk memperoleh *derivatif* dari nilai input. *Rule* dapat mereferensikan sebuah input sebagai aksi *floating* sehingga melewatkannya secara langsung ke *output latch*.

Selama input sampel berikutnya, nilai *output latch* memilih MF nilai *center*, yang berakibat berkurangnya nilai input yang sebelumnya. Beda nilai, dibagi oleh *sampling interval*, adalah nilai *derivatif* yang dapat dijadikan acuan di dalam *rule*.

Sebagai contoh pemakaian input atau aksi di dalam mengukur percepatan motor. *Rule* yang memberikan nilai input ke dalam *output latch* adalah :

VALUE_TO = IN1 if IN1 is MUST_WIN(0, 0, Right Inclusive)

Rule memberikan IN1 sebagai nilai aksi. MUST_WIN adalah tipe *Right Inclusive* mulai nol sehingga apapun nilai IN1, *rule* harus menang dan nilai IN1 diberikan ke *output latch*.

Rule kedua menghitung *derivatif* dan meng-*adjust output drive* ke motor :

ACCEL \pm if IN1 is VALUE_T1 (VALUE_T0, 25, Symmetrical Inclusive)

Maksudnya *rule* menentukan apakah nilai input pada T1 masih di dalam *range* 25 dari nilai awal saat T0. Di dalam aplikasi aktual, perlu MF lain untuk

menentukan polaritas *derivatif* dan *rule* yang lain untuk menjangkau variasi yang lebar.

Contoh di atas *floating membership function* digunakan dengan jelas. Di dalam aplikasinya, *floating MF* dipakai ekstensif untuk menyimpan memori karena lebih sedikit memakai variabel dan *rule* untuk mendeteksi perbedaan input daripada fungsi-fungsi konvensional yang biasa.

2.4.11. Operational Device

Pemrosesan data meliputi beberapa step. Pertama, data sampel analog dikoversikan ke digital dan di-*latch*. Berikutnya *Fuzzifier* membandingkan isi dari *input latch* dengan variabel fuzzy untuk menemukan nilai variabel. *Fuzzifier* juga membentuk penghitungan MAX-of-MIN untuk mencari pemenang *rule*. Terakhir, *Defuzzifier* menentukan pemenang aksi rule dan menahannya untuk konversi ke analog output atau untuk *internal feedback*.

Fuzzifier

Adalah membandingkan data *input latch* dengan MF untuk menghitung nilai fuzzy variabel. Ketika penghitungan MIN *rule* dilakukan, nilainya mewakili *rule* yang disimpan. Ketika penghitungan MAX dilakukan pada seluruh variabel yang mereferensikan nilai output, nilai *rule* pemenang akan diberikan ke *Defuzzifier*.

Peng-update-an Output Latch

Rule dievaluasi dalam urutan saat masuknya. Banyak *rule* dapat mereferensikan output dan output dapat direferensikan berulang-ulang di dalam sebuah *set rule*. Ketika sebuah *rule* atau grup *rule* memberikan output yang dievaluasi dan *rule* selanjutnya memasukkan referensi ke output lain, *compiler* akan menyertakan kode untuk *Last Rule* dengan *output latch* untuk di-update dengan nilai pemenang yang baru. *Latch* data juga bisa dengan cepat dipakai sebagai *feedback*.

Jika setelah pemrosesan *rule* yang berpengaruh ke output lain, *processor* menemukan *rule* atau grup *rule* lain yang menunjuk output sebelumnya, *output latch* akan di-update lagi. Peng-update-an output mungkin bisa sesering mungkin selama proses sebagaimana disana ada bagian grup terpisah yang mereferensikannya.

Sebagaimana sebelumnya, *sampling input* adalah kontinyu. Output Analog juga sering di-update terus menerus. Selama proses variabel *Fuzzy* mungkin memakai *data sample* yang lalu atau dari data yang sedang dipakai proses tergantung dimana *sampling input cycle* berada relatif terhadap *processing cycle*. Jika lebih dari satu grup *rule* yang mereferensi ke input dan output yang sama, maka nilai output akan berubah lebih dari satu kali selama sebuah proses *cycle* berdasar pada perbedaan input data.

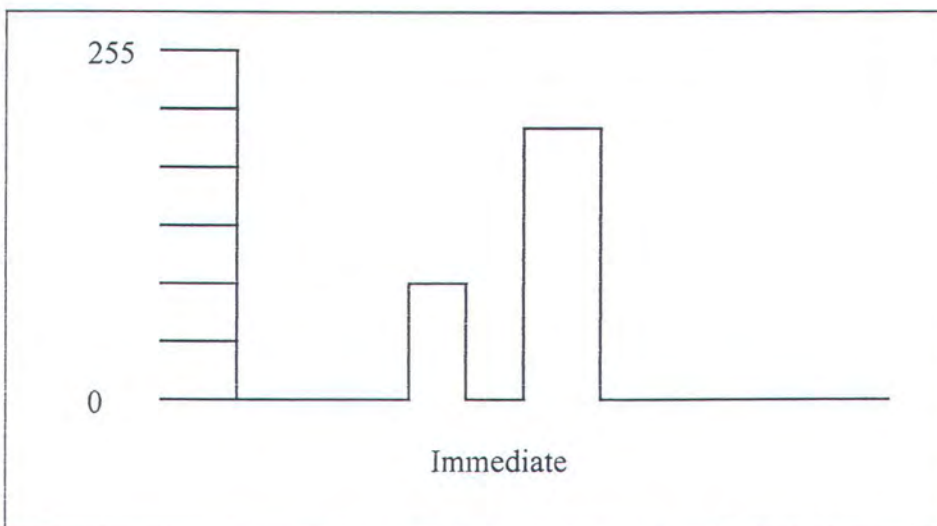
Defuzzifier

Nilai aksi *rule* yang menang dan mode data diberikan ke *defuzzifier*. Data digital dari *defuzzifier* di-latch dan dikonversikan ke analog untuk men-drive output atau diumpanbalikkan kembali secara internal.

Jika semua *rule* dalam sebuah grup mereferensikan sebuah hasil evaluasi output nol, maka grup tidak akan merubah nilainya. Jika lebih dari satu *rule* mengevaluasi dengan hasil nilai paling tinggi dan tidak nol, maka *rule* pertama yang masuk akan menang dan aksinya menentukan output.

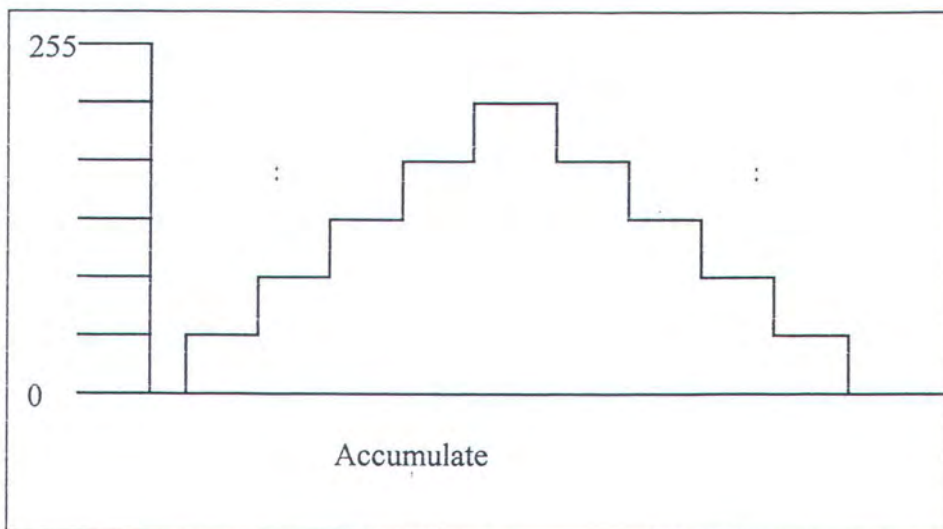
Metode Defuzzifikasi

Hasil *defuzzifikasi* berpengaruh langsung ke output. *Device* ini mempunyai dua metode *defuzzifikasi*, yaitu *Accumulate* dan *Immediate*. Mode *immediate* fungsinya sama dengan tabel, di mana nilai aksi yang menandakan ke *rule* pemenang selama pemasukan, diaplikasikan ke output. *Immediate* dipakai saat nilai output harus *absolute*.



Gambar 2.10. Mode *immediate* defuzzifikasi

Mode *accumulate* adalah untuk menaikkan atau menurunkan nilai output yang ada dengan nilai pemenang *rule*. Output merupakan fungsi dari aksi sekarang dengan aksi sebelumnya. Digunakan pada perubahan output yang halus saat sistem dalam kontrol yang mendekati titik operasinya. Sangat berguna juga pada pembuatan *timing*.



Gambar 2.11. Mode *accumulate* defuzzifikasi

2.5. Fotodioda

Fotodioda adalah merupakan piranti semikonduktor yang dipasang pada *window* ke *junction* sehingga jika ada cahaya yang mengenainya membuat pasangan-pasangan hole dan elektron akan mengalirkan arus. Fotodioda yang terpasang pada gambar 2.12. terlihat bahwa ujung anoda dihubungkan ke input terminal negatif dari op amp sedangkan katoda diberi tegangan input (V_{in}) sehingga menyebabkan dioda reversed bias. Dalam kondisi gelap fotodioda masih

mengalirkan sejumlah kecil arus bocor pada orde nano ampere. Akan tetapi kerja dari fotodioda tersebut tergantung dari energi cahaya yang mengenainya, sehingga akan mengalirkan arus $50\text{ }\mu\text{A}$ atau lebih. Karena itu arus yang mengalir hanya tergantung dari energi cahaya yang mengenai dan selanjutnya arus tersebut diubah menjadi tegangan oleh tahanan *feedback* (R_f) yang dapat dinyatakan dalam bentuk $V_{out}=V_{Rf}=I.R_f$. Fotodioda bekerja berdasarkan efek fotokonduksi sehingga mempunyai puncak spektrum untuk panjang gelombang yang berbeda-beda, tergantung dari tipenya. Oleh karena itu fotodioda mampu melewatkan panjang gelombang cahaya yang mengenainya sesuai dengan spesifikasinya dan memproteksi panjang gelombang yang berada di luar jangkauannya. Demikian halnya dengan fototransistor. Hanya saja, pada fototransistor cahaya yang mengenai junction basis emitor dikuatkan untuk memberikan arus kolektor yang besar. Jadi fotodioda dan fototransistor dapat bekerja dengan waktu jangkit yang relatif singkat.

BAB III

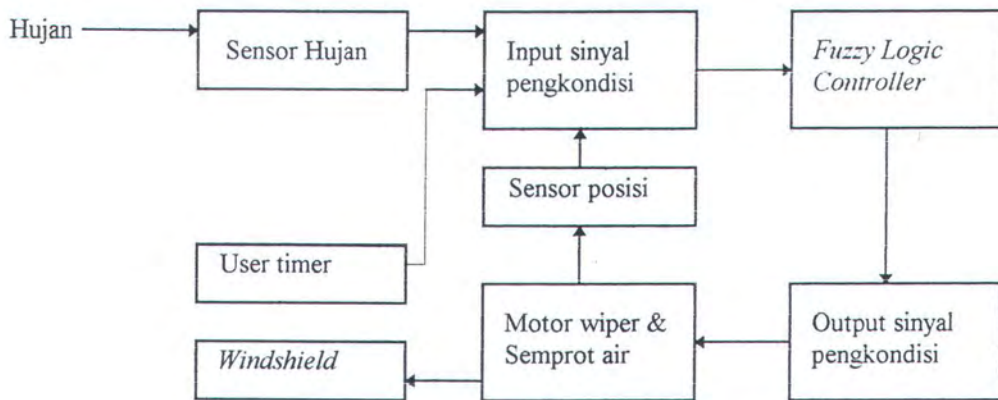
PEMBUATAN PERANGKAT LUNAK DAN PERANGKAT KERAS

Sistematika dalam pembuatan perangkat keras :

1. Membuat diagram blok secara keseluruhan dan cara kerja umum.
2. Membuat rangkaian elektronika untuk setiap blok sesuai dengan kebutuhannya.

3.1. PEMBUATAN SISTEM

Diagram blok dari sistem yang direncanakan dapat dilihat pada Gambar 3.1. dimana terdapat tiga blok utama yakni bagian *Fuzzy Logic*, blok sensor (yang meliputi sensor kondisi hujan, sensor intensitas hujan, dan sensor posisi wiper), serta blok *user timer*. Cara kerja umum dari tugas akhir ini adalah apabila blok sensor aktif karena ada respon dari luar, maka hasil data yang didapat akan diolah oleh blok *Fuzzy Logic* untuk memberikan aksi atau respon berupa pengaturan kecepatan motor wiper atau semprotan air. Pada saat sensor menangkap kondisi hujan basah dan intensitas hujan dalam keadaan gerimis maka *Fuzzy Logic* akan memberi sinyal kondisi kepada motor wiper agar bekerja pada kecepatan lambat, tetapi apabila pada kondisi hujan kering dan intensitas hujan gerimis maka *Fuzzy*



Gambar 3.1. Blok diagram sistem

Fungsi dari masing-masing blok adalah sebagai berikut :

1. Blok sensor

Sensor kondisi hujan berfungsi untuk membedakan apakah hujan dalam keadaan kering, basah, atau sangat basah. Sensor intensitas hujan berfungsi untuk membedakan apakah hujan dalam keadaan gerimis, deras, atau sangat deras.

Sensor posisi berfungsi untuk mengamati posisi wiper apakah sudah berada pada posisi yang ditentukan atau belum.

2. Blok *Fuzzy Logic* berfungsi untuk mengatur *switch* kecepatan motor, menganalisa input yang masuk dan memberikan respon sesuai dengan *rule-rule* yang dibuat.
3. Blok *User Timer* berfungsi untuk proses pembersihan kaca mobil secara otomatis dan dalam waktu tertentu.

3.2. PEMBUATAN PERANGKAT KERAS

Sebagai langkah awal dilakukan pemilihan rangkaian dan komponen yang mudah dalam perencanaan maupun pencarian komponen di pasaran. Sesuai dengan fungsinya, perangkat keras yang direncanakan sebagai berikut :

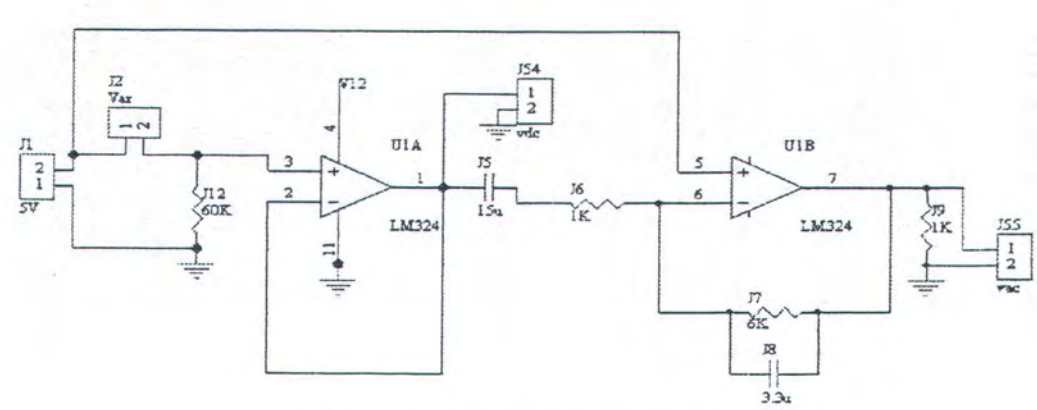
- Sensor hujan (intensitas dan kondisi kebasahan)
- Sensor posisi
- *User timer*
- *Driver* motor dan semprot air
- Display kerja sistem
- Modul NLX220

3.2.1. Sensor Hujan

Perancangan rangkaian sensor hujan diharapkan dapat mendeteksi intensitas hujan dan kondisi kebasahan hujan. Karena pendeteksian ini tidak diperlukan adanya tingkat ketelitian yang tinggi maka sensor dirancang secara sederhana.

Pada prinsipnya sensor kondisi hujan menggunakan metode konduktivitas atau penghantar dimana dua buah penghantar diletakkan pada jarak tertentu dengan medium udara yang mula-mula mempunyai harga resistansi tak berhingga. Apabila terjadi hujan yang semakin deras maka butir-butir hujan akan lebih banyak menyentuh kedua celah penghantar dan akibatnya akan terjadi perubahan resistansi yang semakin kecil. Agar perubahan kondisi hujan sejalan dengan perubahan tegangan maka digunakan metode pembagi tegangan.

Untuk sensor intensitas hujan digunakan metode *band pass filter* dimana akan menyeleksi daerah frekuensi yang beroperasi. Air hujan yang menembus kedua celah penghantar berupa gerakan butir-butir air yang mempunyai frekuensi-frekuensi rendah. Frekuensi ini berkisar antara 1-10Hz.³



Gambar 3.2. Rangkaian sensor hujan

Keterangan :

- Nilai Var berkisar antara 20kΩ sampai 50kΩ dan diusahakan agar input pada pin Vdc mempunyai tegangan antara 0 sampai 5 Volt.

$$Vo = \frac{R}{(R + 20k)} \cdot Vi$$
 , sehingga didapatkan harga R sekitar 60kΩ.

- Untuk rangkaian *band pass filter* diambil dari buku referensi.

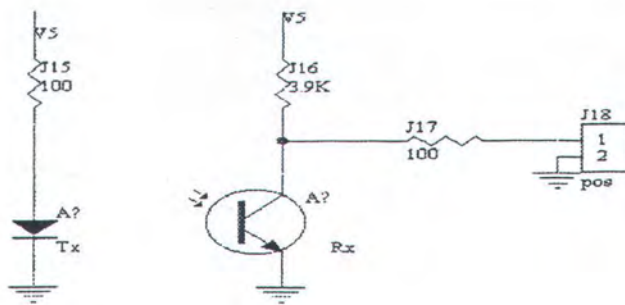
3.2.2. Sensor Posisi

Perancangan rangkaian sensor posisi diharapkan dapat mendeteksi apakah posisi wiper sudah pada tempat referensi atau belum. Pada tugas akhir ini

³ -----, IEEE Control System halaman 31

digunakan *optocoupler* yang karakteristiknya sudah dijelaskan pada bab sebelumnya.

Setiap kali wiper melewati sensor maka akan terjadi pemutusan arus I_b yang menyebabkan tegangan outputnya sekitar 4 Volt, sedangkan apabila tidak melewati maka akan menyebabkan tegangan outputnya 0 Volt.



Gambar 3.3. Rangkaian sensor posisi

3.2.3. User Timer

Pada perancangan *user timer* ini digunakan *timer IC* LM555 yang dirancang untuk tipe *monostable multivibrator*. Saat selektor mengacu pada saklar 1 maka akan bekerja *delay* sampai selang waktu 1 detik. Pada saklar 2 akan bekerja *delay* sampai selang waktu 10 detik. Demikian pula untuk saklar-saklar yang lainnya.

Untuk aplikasi yang sebenarnya, *user timer* dapat diset sampai beberapa jam sesuai dengan kemauan pemakai.

Hasil dari IC 555 ini diumpankan ke rangkaian *schmitt trigger* yang juga berfungsi sebagai *delay*. Pada prinsipnya rangkaian *schmitt trigger* ini merupakan

rangkaian *convertor* yang outputnya adalah kebalikan dari inputnya. Selain itu dengan rangkaian ini dapat diatur pula waktu aktifnya dengan mengubah parameter R dan C.

Persamaan untuk menentukan waktu aktif rangkaian ini adalah sebagai berikut :

$$t = RC \ln \frac{V_{DD}}{(V_{DD} - V_P)}$$

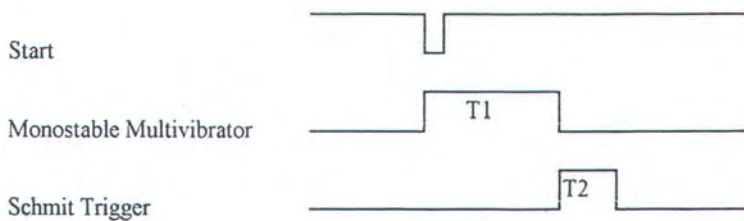
dimana : R = tahanan (ohm)

C = kapasitor (farad)

V_{DD} = Tegangan Drain (volt)

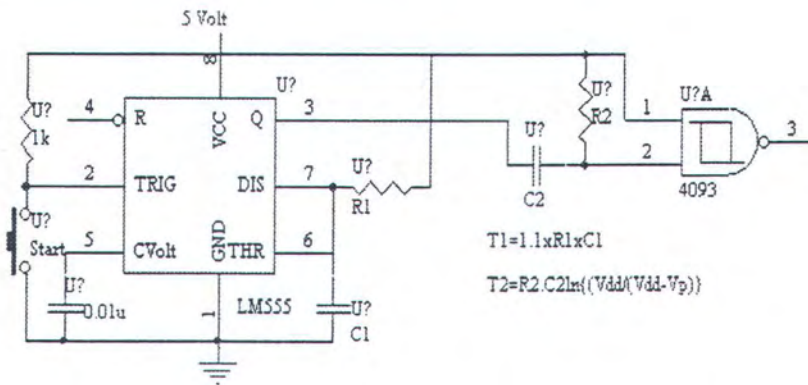
V_P = Tegangan positif (volt)

Perbedaan fungsi kedua blok ini dapat dijelaskan melalui *timing diagram* sebagai berikut :



Dari *timing diagram* di atas tampak bahwa T1 adalah selang waktu yang diinginkan pemakai untuk menjalankan operasi pembersihan kaca secara otomatis, sedangkan T2 merupakan waktu yang diperlukan bagi gerakan wiper dan semprotan air untuk menjalankan prosesnya.

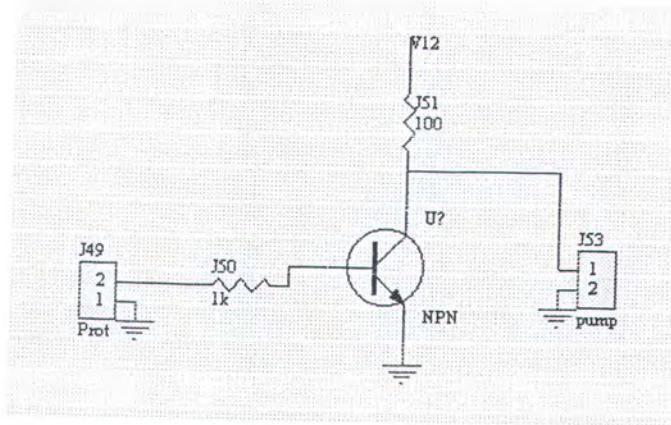
Hasil dari *user timer* ini digunakan sebagai input *Fuzzy Logic* yang di dalam *rule*-nya menghasilkan output berupa pembersihan kaca dengan menyemprotkan air dan gerakan tangkai wiper.



Gambar 3.4. Rangkaian *User Timer*

3.2.4. *Driver Motor Wiper dan Semprot Air*

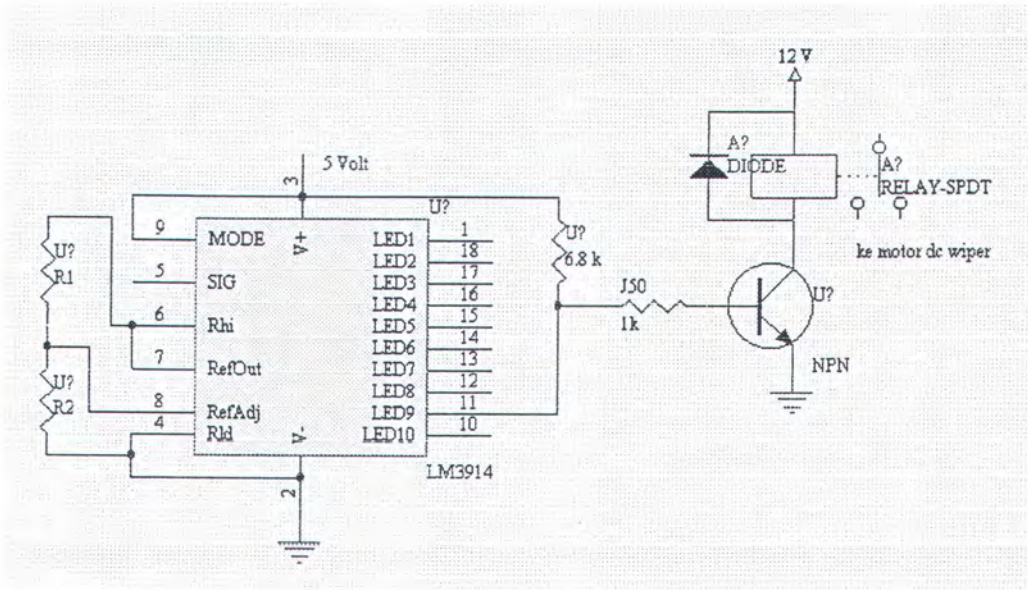
Untuk *driver* semprot air digunakan fungsi transistor sebagai saklar on-off dimana apabila pada transistor diberikan tegangan atau arus pada terminal basisnya maka tegangan outputnya akan 0 Volt. Tetapi apabila tidak diberikan tegangan atau arus pada tegangan basisnya maka tegangan outputnya akan bernilai sesuai dengan tegangan Vcc yang mampu men-*drive* motor dc penyemprot.



Gambar 3.5. Rangkaian *driver* semprot air

Untuk *driver* motor wiper selain menggunakan prinsip transistor sebagai saklar *on-off* juga digunakan rangkaian penggerak display led (LM3914) yang fungsi utamanya adalah untuk menyeleksi input yang masuk. Pada perangkat keras motor dc untuk wiper sudah terdapat terminal-terminal untuk mengatur kecepatan (terdapat tiga terminal yaitu terminal ground, terminal kecepatan 1, dan terminal kecepatan 2) sehingga dalam pengaturan kecepatannya cukup menggunakan sistem *switching* kecepatan 1 atau 2.

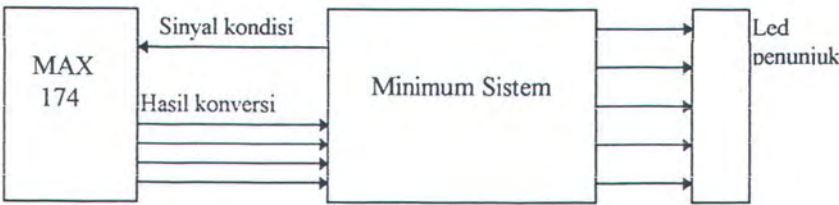
Misalnya pada output dari modul *Fuzzy* mengeluarkan tegangan 5 Volt maka output dari IC LM3914 *pin* yang ke-11 aktif (*active low*) yang kemudian akan diumpankan ke transistor untuk mengaktifkan terminal kecepatan 2.



Gambar 3.6. Rangkaian *driver* motor dc wiper

3.2.5. Display Kerja Sistem

Modul ini berfungsi sebagai penampil keadaan hujan yang sedang terjadi. Terdapat dua bagian utama yaitu pengubah A/D dan *minimum system*. Pengubah A/D yaitu untuk mengubah input yang masuk berupa perubahan tegangan dari kondisi dan intensitas hujan ke dalam bentuk digital. Sedangkan *minimum system* digunakan sebagai sinyal pengkondisi untuk ADC sekaligus mengolah sinyal digital yang dihasilkan ADC menjadi tampilan penunjuk keadaan hujan.



Gambar 3.7. Blok diagram display unjuk kerja sistem

3.3. PEMBUATAN PERANGKAT LUNAK

Perangkat penunjang aplikasi ini adalah *Insight* yang digunakan untuk mendefinisikan :

1. Input
2. Output
3. Variabel *Fuzzy*
4. *Rules*
5. Simulasi dan *Down load rule* ke IC

3.3.1. Input

Penentuan definisi input ditentukan berdasarkan kebutuhan. Dalam hal ini terdapat input internal dan input eksternal. Input internal merupakan umpan balik internal yang secara fisik tidak dibutuhkan hubungan ke komponen luar, namun secara internal dihubungkan dengan *software (loop back)*. Sedangkan input eksternal harus dihubungkan secara fisik.

Untuk jenis input yang berhubungan langsung dengan terminal output sebagai umpan balik internal adalah count yang berfungsi untuk mengatur *delay* untuk kondisi tertentu.

Sedangkan yang dihubungkan secara perangkat keras dengan input adalah sebagai berikut :

1. Kondisi, berfungsi untuk mengetahui kondisi hujan (tingkat kebasahan).
2. Intensif, berfungsi untuk mengetahui intensitas hujan.

3. Posisi, berfungsi untuk mengetahui posisi wiper pada titik referensi.
4. Timer, berfungsi untuk mengatur waktu pembersihan kaca mobil secara otomatis dalam waktu tertentu.

3.3.2. Output

Output yang langsung dihubungkan dengan komponen di luar adalah :

1. Motordc, besarnya tegangan yang dihubungkan dengan motor dc dari wiper.
2. Semprot, besarnya tegangan yang dihubungkan dengan motor dc dari semprotan mobil.

3.3.3. Variabel *Fuzzy*

Pembagian variabel *fuzzy* dibagi menjadi :

1. Deteksi kondisi kebasahan hujan
 - Kondisi is kering+ (50, 20, Left Inclusive)
 - Kondisi is basah (165, 18, Symmetric Inclusive)
 - Kondisi is basah+ (200, 23, Right Inclusive)
 - Kondisi is ker_bas (138, 18, Symmetric Inclusive)
 - Kondisi is kering(95, 30, Symmetric Inclusive)
2. Deteksi intensitas hujan
 - Intensif is deras+ (60, 35, Left Inclusive)
 - Intensif is deras (120, 40, Symmetric Inclusive)
 - Intensif is no_rain (245, 12, Right Inclusive)

- Intensif is rintik (220, 17, Symmetric Inclusive)
- Intensif is gerimis (180,30, Symmetric Inclusive)

3. Deteksi posisi wiper

- Posisi is nonaktif (54, 0, Left Inclusive)
- Posisi is aktif (55, 0, Right Inclusive)

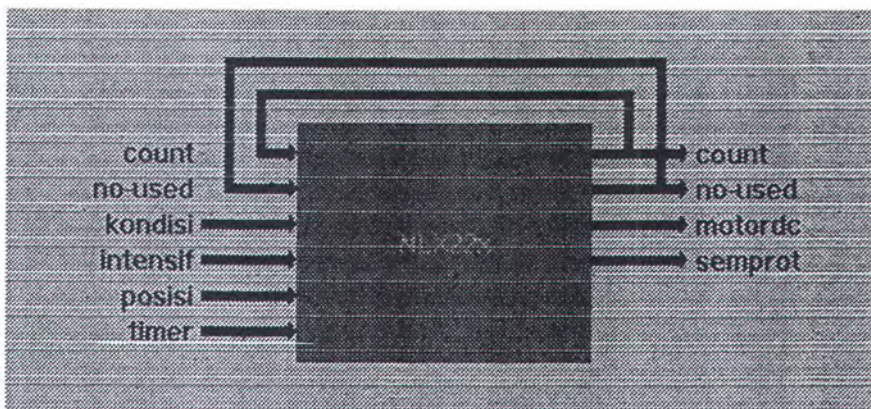
4. User timer

- Timer is disable (63, 0, Left Inclusive)
- Timer is aktif (0, 0, Right Inclusive)

5. Counter

- Count is min (150, 0, Symmetric Exclusive)
- Count is max (150, 12, Symmetric Inclusive)

3.3.4. Rules



Detail *rule* adalah sebagai berikut :

If count is min then count+1

If count is max then count=0

If kondisi is kering+ and intensif is no_rain and posisi is aktif then
motordc=0

If kondisi is kering+ and intensif is rintik and posisi is aktif then
motordc=0

If kondisi is kering+ and intensif is gerimis and posisi is aktif then
motordc=0

If kondisi is kering+ and intensif is deras and posisi is aktif then
motordc=0

If kondisi is kering+ and intensif is deras+ and count is max then
motordc=128

If kondisi is kering and intensif is no_rain and posisi is aktif then
motordc=0

If kondisi is kering and intensif is rintik and posisi is aktif then motordc=0

If kondisi is kering and intensif is gerimis and posisi is aktif then
motordc=0

If kondisi is kering and intensif is deras and count is max then
motordc=128

If kondisi is kering and intensif is deras+ then motordc=128

If kondisi is ker_bas and intensif is no_rain and posisi is aktif then
motordc=0

If kondisi is ker_bas and intensif is rintik and posisi is aktif then
motordc=0

If kondisi is ker_bas and intensif is gerimis and count is max then
motordc=128

If kondisi is ker_bas and intensif is deras then motordc=128

If kondisi is ker_bas and intensif is deras+ and count is max then
motordc=255

If kondisi is basah and intensif is no_rain and posisi is aktif then
motordc=0

If kondisi is basah and intensif is rintik and posisi is aktif then motordc=0

If kondisi is basah and intensif is gerimis then motordc=128

If kondisi is basah and intensif is deras and count is max then
motordc=255

If kondisi is basah and intensif is deras+ then motordc=255

If kondisi is basah+ and intensif is no_rain and posisi is aktif then
motordc=0

If kondisi is basah+ and intensif is rintik and count is max then
motordc=128

If kondisi is basah+ and intensif is gerimis then motordc=128

If kondisi is basah+ and intensif is deras and count is max then
motordc=255

If kondisi is basah+ and intensif is deras+ then motordc=255

If kondisi is basah+ and intensif is deras+ then semprot=0

If timer is disable then semprot=200

If timer is aktif then semprot=0

If timer is aktif then motordc=255

If timer is disable and posisi is aktif then motordc=0

3.3.5. Simulasi dan *Download rule* ke IC

Simulasi dilakukan dengan fasilitas yang ada pada *software insight* yaitu dengan membuka *file* dengan ekstension .ins dan .inp.

Download rule ke IC dilakukan dengan mengubah file dengan ekstension .ins menjadi *file* dengan ekstension .220 dengan format bin. Setelah itu *file* dengan ekstension .220 di-convert menjadi *file* dengan ekstension .bin yang siap untuk diisikan ke IC.

BAB IV

PENGUKURAN DAN PENGUJIAN

Pada bab ini akan dibahas mengenai pengujian alat yang dilanjutkan dengan pengukuran dan kalibrasi masing-masing bagian perangkat keras yang telah dibuat dengan alat ukur yang tersedia. Pengujian dan pengukuran dilakukan secara bertahap yakni dimulai dari blok sensor, blok *user timer*, serta blok *Fuzzy Logic*.

4.1. Pengukuran Modul

Pengukuran dilakukan untuk mengetahui apakah alat yang telah dibuat sudah bekerja dengan baik atau belum dan apakah sudah sesuai dengan yang direncanakan. Pengukuran dilakukan pada masing-masing modul yang telah dibuat yaitu :

- Sensor hujan (intensitas dan kondisi kebasahan)
- Sensor posisi
- *User timer*
- *Driver* motor dan semprot air
- Modul NLX220

4.1.1. Modul Sensor Hujan (Intensitas dan Kondisi Kebasahan)

Untuk mengukur tanggapan sensor ini yaitu dengan memberi tegangan DC sebesar +12 Volt pada *pin* V12 dan menghubungkan sensor hujan pada *pin* VAR. Sedangkan untuk simulasi hujan digunakan semprotan air.

Pada pengukuran tanggapan sensor dilakukan dengan mensimulasi hujan dengan semprotan air, dimana hasil intensitas hujan pada *pin* Vac dan kondisi hujan pada *pin* Vdc sebagai berikut :

a. Intensitas hujan

- Tidak hujan = 9 - 10 Volt
- Rintik-rintik = 8 - 9 Volt
- Gerimis = 6 - 8.5 Volt
- Deras = 3 - 6.5 Volt
- Deras sekali = 0 - 3.5 Volt

b. Kebasahan hujan

- Sangat kering = 0 - 1.75 Volt
- Kering = 1.5 - 2.75 Volt
- Kering kebasahan = 2.5 - 3.15 Volt
- Basah = 3.0 - 3.75 Volt
- Basah sekali = 3.5 - 5.0 Volt

4.1.2. Modul Sensor Posisi

Langkah pengukuran pada rangkaian ini dilakukan dengan menutup (aktif) atau membuka (non aktif) dengan suatu penghalang. *Pin* Tx dihubungkan dengan fotodioda pemancar dan *pin* Rx dihubungkan dengan fotodioda penerima. Sedangkan untuk hasil proses pada *pin* POS.

Hasil yang didapat yaitu pada saat aktif, tegangan pada *pin* POS sekitar 2 sampai 5 Volt (tergantung pada posisi penghalang) dan pada saat non aktif, tegangannya akan nol.

4.1.3. Modul User Timer

Pengukuran waktu pada rangkaian *user timer* dilakukan dengan cara memilih saklar interval waktu pada IC 555. Saklar pertama menghasilkan waktu tunda selama 2 detik, saklar kedua menghasilkan waktu tunda selama 10 detik, dan saklar ketiga menghasilkan waktu tunda selama 30 detik.

Hasil dari IC 555 ini diumpankan ke rangkaian *schmit trigger* untuk mengaktifkan gerakan motor wiper dan semprot selama 3 detik.

4.1.4. Driver Motor dan Semprot Air

Driver motor wiper terbagi atas dua bagian yaitu rangkaian seleksi tegangan (LM3914) dan rangkain transistor sebagai saklar. Rangkaian seleksi tegangan diuji dengan memberikan tegangan pada *pin* DIFF untuk dilakukan seleksi apakah inputnya 0, 2.5, atau 5 Volt. Dalam pengukuran diberi tegangan input sebesar 2.5 Volt maka hanya satu yang mempunyai hasil nol, sedangkan *pin*

yang lainnya mempunyai tegangan sekitar 4 Volt. Hasil keluaran rangkaian seleksi tegangan selanjutnya diproses oleh rangkaian transistor sebagai saklar dimana hanya tegangan masukan nol saja yang menghasilkan output sebesar 12 Volt untuk men-*drive* motor dc wiper.

Untuk pengukuran *driver* semprot air, *pin* PROT diberi tegangan 0 dan 4 Volt. Pada tegangan 0 Volt, *pin* PUMP mempunyai tegangan 12 Volt yang mampu men-*drive* motor penyemprot. Sedangkan pada tegangan 4 Volt, *pin* PUMP mempunyai tegangan 0 Volt dan akibatnya motor penyemprot tidak aktif.

4.1.6. Modul NLX220

Untuk menguji modul ini, dibuat *rule* atau aturan yang sederhana yaitu harga tegangan input sama dengan harga tegangan output. Sehingga dari hal ini dapat mengetahui apakah modul NLX220 sudah berjalan atau belum. Selain itu dapat juga diketahui urutan *pin* input dan *pin* output sehingga mempermudah pengujian sistem.

Variabel *Fuzzy* yang digunakan adalah sebagai berikut :

- input is win1 (10,0,right inclusive)
- input2 is win2(0,0,right inclusive)
- input3 is win3(0,0,right inclusive)
- input4 is win4(0,0,right inclusive)

Sedangkan *rule-rule*-nya sebagai berikut :

- if input1 is win1 then output1=input1

- if input2 is win2 then output2=input2
- if input3 is win3 then output3=input3
- if input4 is win4 then output4=input4

Dengan menggunakan kelompok *rule* tersebut maka pada setiap input diberi sumber tegangan dan outputnya diukur. Dari pengukuran diperoleh hasil bahwa harga output sama dengan hasil input. Hal ini membuktikan bahwa modul telah bekerja dengan baik.

4.2. Pengujian Sistem

Setelah modul-modul tersebut di atas diuji, langkah selanjutnya adalah menguji secara keseluruhan.

BAB V PENUTUP

5.1. Kesimpulan

Dari hasil perencanaan dan pembuatan alat serta hasil pengukuran dan pengujian sistem secara keseluruhan dapat disimpulkan sebagai berikut :

1. Penggunaan teknologi *Fuzzy Logic* dapat diterapkan pada optimasi penggunaan *windshield wiper* secara otomatis karena memiliki beberapa keunggulan antara lain :
 - Tidak memerlukan model matematika yang rumit sehingga mempermudah proses perancangan.
 - Aturan *fuzzy* bersifat intuitif sehingga mudah dimengerti.
 - Memiliki fleksibilitas tinggi karena mudah untuk mengubah aturan atau *rule* sesuai dengan keperluan.
2. Pengaturan gerakan wiper dan aktifitas semprotan air menjadi lebih mudah dan praktis dengan menggunakan alat yang dibuat meskipun tidak menghilangkan unsur aslinya yaitu secara manual.
3. Dalam tugas akhir ini telah diuji bagaimana *fuzzy logic* dapat diaplikasikan sebagai pengontrol *windshield wiper system* dengan menggunakan sensor konduktivitas.

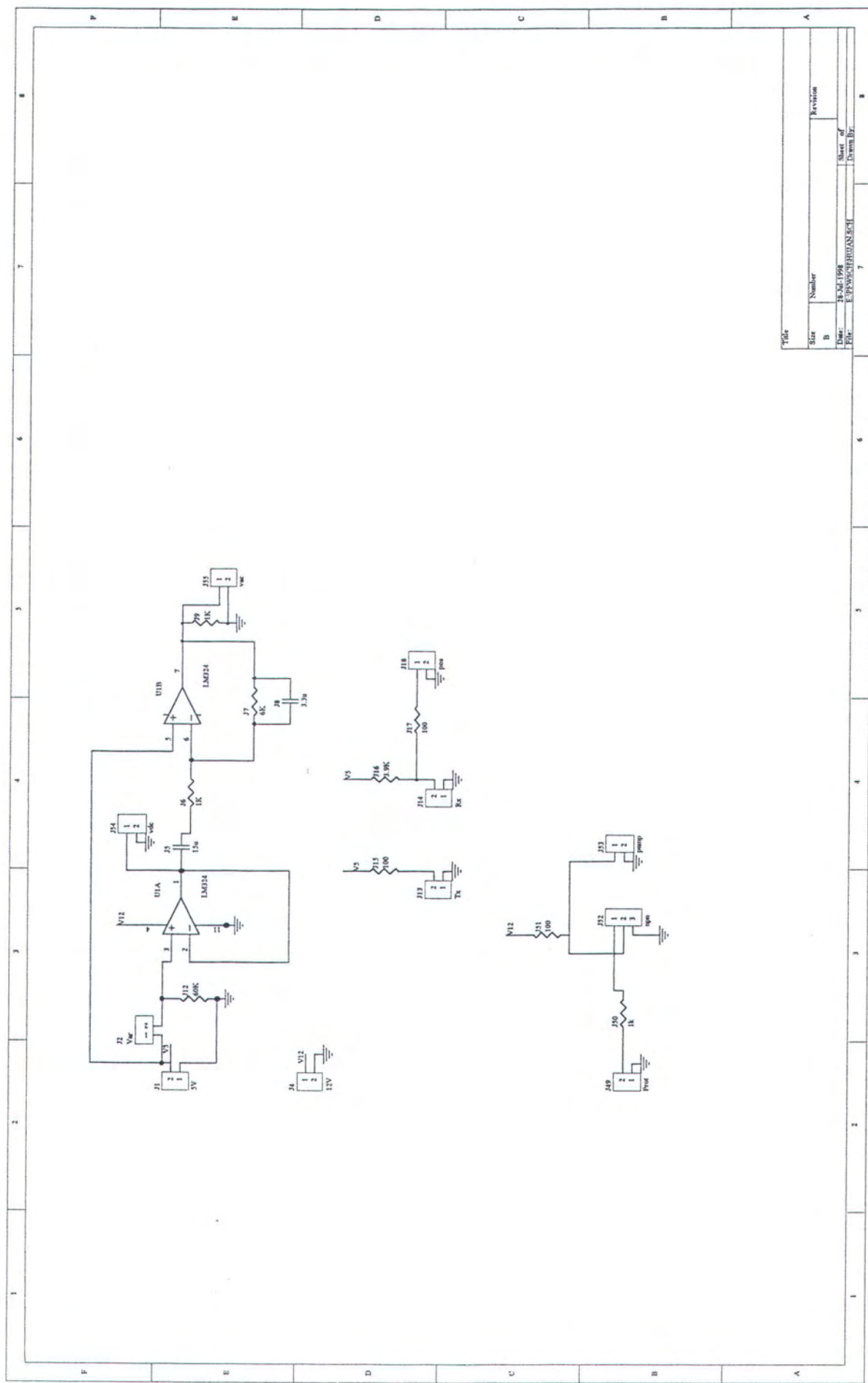
5.2. Saran

Berdasarkan hasil pengujian alat, terdapat beberapa hal yang bisa diperbaiki.

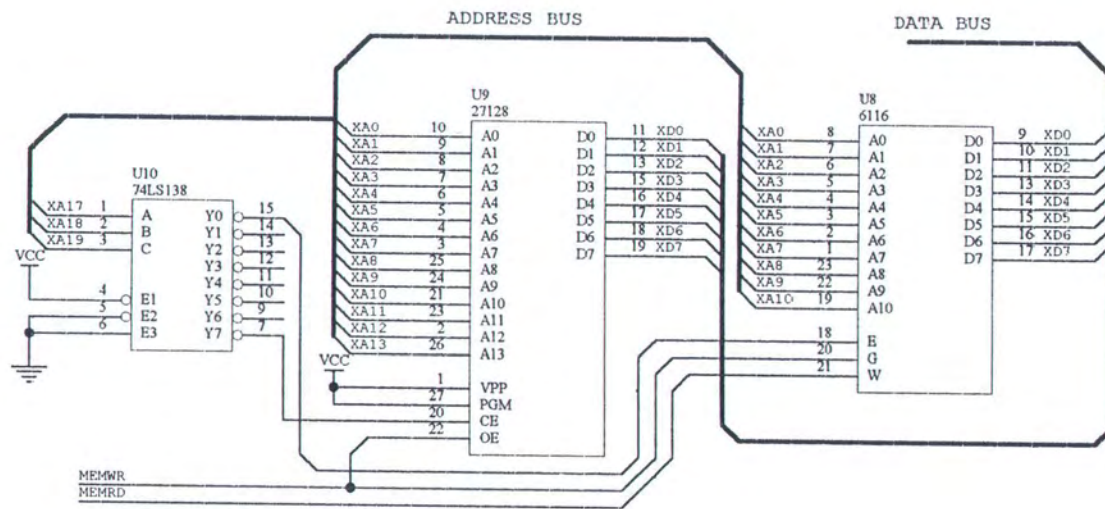
1. Sensor hujan yang digunakan sebaiknya berbentuk pipih supaya bila terkena sapuan *windshield wiper*, kekeringan sensor sesuai dengan kekeringan kaca mobil.
2. Untuk memperbaiki unjuk kerja sensor hujan perlu diberikan *buffer* (penyangga) dan filter.
3. Kecepatan motor wiper supaya lebih adaptif dengan kondisi dan intensitas hujan dapat ditambahkan pengaturan arus motor yang otomatis.

DAFTAR PUSTAKA

1. Constantin von Altrock, 1995, *Fuzzy Logic and Neuro Fuzzy Applications Explained*, New Jersey: Prentice Hall.
2. Fredrick W. Hughes, PANDUAN OPERASIONAL AMPLIFIER, Penerbit PT ELEX MEDIA KOMPUTINDO Kelompok Gramedia Jakarta.
3. Robert F. Coughlin and Fredirick F Driscoll, 1982, Penguat Operasional dan Rangkaian Terpadu Linear edisi kedua, Prentice Hall, Inc
4. Roger L. Tokheim, 1990, *Digital Electronics second edition*, McGraw-Hill, Inc.
5. Ramakant A. Gayakwad, *Op-Amps And Linear Integrated Circuit*, Second Edition, Prentice Hall Of India Private Limited, New Delhi, September, 1988.
6. ----1991, DATA SHEET BOOK I, Diterjemahkan oleh Wasito S, Jakarta: Penerbit PT Elex Media Computindo-Kelompok Gramedia-
7. -----1994. PIC16C5X, *EPROM-Based 8-bit CMOS Microcontroller Series*. Microchip Technology Inc.
8. -----1998, *NEW RELEASES DATA BOOK*, California: Maxim Integrated Products.
9. -----, *IEEE Control Systems*.
10. -----, *Stand Alone Fuzzy Logic Controller NLX220*, Adaftive Logic



Title		Revision	
Size	Number	Revision	
B			
Date	14.04.1999	Sheet of	7
File	EXPNSCHTUNANGEN	Drawn by	



1

2

3

4

D

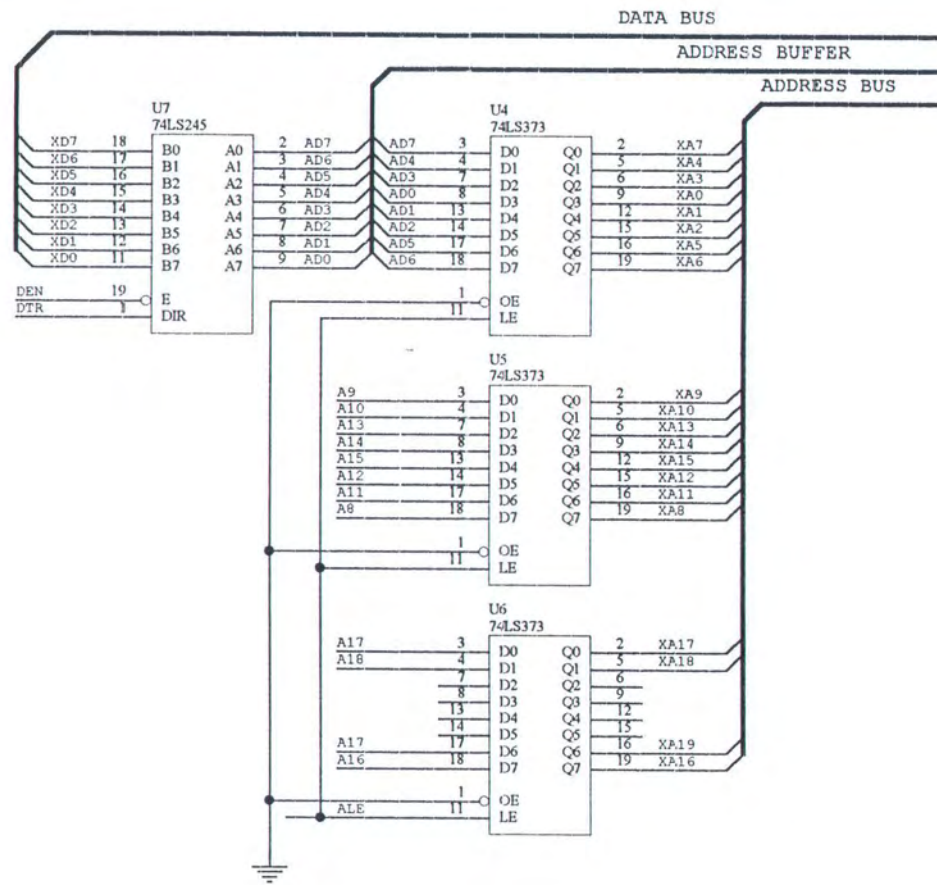
C

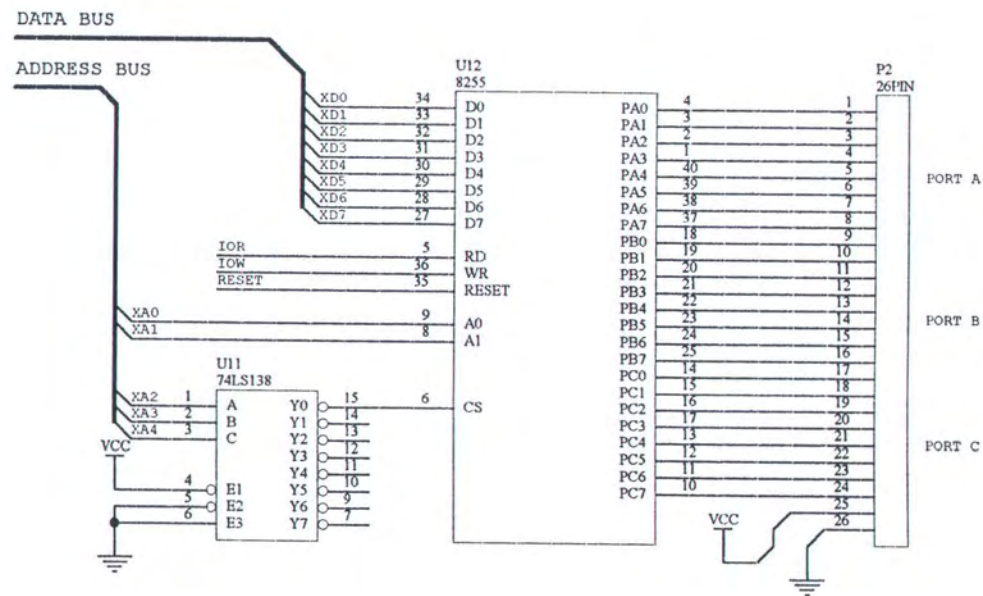
B

D

C

B





BIODATA



Zudha Nooradillah dilahirkan di Surabaya, Jawa Timur pada tanggal 15 April 1974 dari pasangan Djarwadi Alrasyid dan Marmiyah yang saat ini bertempat tinggal di Pondok Marinir I-10 Masangan Kulon Sukodono Sidoarjo dan merupakan putra kedua dari tiga bersaudara.

Riwayat pendidikan penulis adalah sebagai berikut :

Tahun 1985 : Lulus dari SD Hang Tuah VIII Surabaya

Tahun 1990 : Lulus dari SMP Negeri 24 Surabaya

Tahun 1993 : Lulus dari SMA Negeri 5 Surabaya

Tahun 1993 : Diterima di Jurusan Teknik Elektro FTI-ITS dengan nrp. 2293100001 melalui program UMPTN

Pada awal bulan Agustus 1998 mengikuti seminar dan ujian Tugas Akhir pada Bidang Studi Elektronika dan diharapkan dapat mengikuti wisuda pada akhir bulan Agustus 1998.

Selama kuliah di Jurusan Teknik Elektro FTI-ITS, penulis aktif dalam kegiatan kemahasiswaan antara lain :

- Pengurus divisi workshop pada Himpunan Mahasiswa Teknik Elektro FTI-ITS tahun 1996-1997
- Koordinator Laboratorium Rangkaian Diskrit Bidang Studi Elektronika tahun 1997-1998
- Koordinator praktikum Elektronika untuk mahasiswa dari STTAL dan D3 Komputer Kontrol
- Asisten praktikum Rangkaian Listrik, Elektronika, dan Elektronika lanjut I di Laboratorium Bidang Studi Elektronika